EINFÜHRUNG IN DIE BENUTZUNG VON SPSS VON DIETMAR SCHRAUSSER

INHALT:

1 ALLGEMEINE BEMERKUNGEN ZUR COMPUTERUNTERSTÜTZTEN DATENANALYSE 3

1.1 SPRACHEN 3
1.1.1 Programmiersprachen 3
1.1.2 Betriebssystemsprachen 3
1.1.3 Programmsysteme 3
1.2 BENUTZERINTERAKTION 4
1.2.1 Der Interaktive Betrieb 4
1.2.2 Der 'Batch' Betrieb 4
1.3 FAUSTREGELN 4

2 DAS GRUNDPRINZIP DES PROGRAMMSYSTEMS SPSS 5

2.1 ALLGEMEINE STRUKTUR DER ARBEITSWEISE UND DIE 3 GRUNDLEGENDEN SCHRITTE 5
2.2 DATENSTRUKTUR 6
2.3 SYNTAXSTRUKTUR 6
2.4 SPSS AUF DER VAX 6
2.4.1 Vorteile 6
2.4.2 Nachteile 7
2.5 SPSS UNTER WINDOWS 7
2.5.1 Vorteile 7
2.5.2 Nachteile 7

3 SPSS AUF DER VAX SPSSX 7

3.1 VMS VERZEICHNISSTRUKTUR 7
3.2 VMS DATEINOTATION 8
3.3 VMS STARTEN 8
3.4 VMS BEFEHLE 9
3.4.1 Verzeichnis erstellen 9
3.4.2 Verzeichnis wechseln 9
3.4.3 Editor-Befehle 10
3.4.3.1 Dateien erzeugen 10
3.4.3.2 Editor beenden, Dateien speichern 11
3.4.4 Dateien löschen 11
3.4.5 Verzeichnis und Dateien auflisten 12
3.4.6 Aussteigen 12
3.5 SPSSX 13
3.5.1 Dateneingabe 13
3.5.2 Syntaxeingabe 13
3.5.3 Berechnung 14
3.5.4 Ergebnis 15
3.5.5 Weitere Möglichkeiten 16

4 SPSS UNTER WINDOWS SPSSWIN 16

4.1 VERZEICHNISSTRUKTUR 16
4.2 DATEINOTATION 16

4.3 STARTEN 17
4.4 GRUNDLEGENDE 'MOUSECLICKS' 17
4.5 SPSSWIN 18
4.5.1 Starten 18
4.5.2 Dateneingabe 20
4.5.3 Syntaxeingabe 21
4.5.4 Berechnung 22
4.5.5 Ergebnis 22
4.5.6 Aussteigen 23

5 BEISPIELE 24

6 SPSS-BEFEHLE UND PROZEDUREN 24

6.1 ALLGEMEIN 24
6.2 ROHDATEN EINLESEN 25
6.2.1 ASCII-Daten einlesen 25
6.2.2 Binär-Daten einlesen 25
6.2.3 Direkte Dateneingabe 26
6.3 ROHDATEN SPEICHERN 26
6.3.1 Als ASCII-Datei speichern 26
6.3.2 Als Binär-Datei speichern 27
6.4 ROHDATENMANIPULATION 27
6.4.1 Berechnung neuer Variablen 27
6.4.2 Bedingte Berechnung neuer Variablen 28
6.4.3 Schleifenbildung 28
6.4.4 Umcodieren einzelner Werte einer Variablen 29
6.4.5 Bedingte Auswertung 29
6.4.6 Auswahl einer gewissen Anzahl von Fällen 30
6.4.7 Auf eine Prozedur beschränkte Auswahl 30
6.4.8 Rohdaten sortieren 31
6.4.9 Aufteilung der Rohdatenmatrix 31
6.4.10 Transponierung der Rohdatenmatrix 32
6.5 GESTALTUNG DER AUSGABEDATEI 32
6.5.1 Setzen von globalen Eigenschaften 32
6.5.2 Titel in der Ausgabedatei 33
6.5.3 Beschreibung von Variablen 33
6.5.4 Beschreibung einzelner Werte 33
6.6 STATISTIKPROZEDUREN 34
6.6.1 Deskriptive Statistik 34
6.6.1.1 Auflisten von Variablen 34
6.6.1.2 Häufigkeitstabellen 34
6.6.1.3 Kreuztabulation 35
6.6.1.4 Regressionsgraph 36
6.6.1.5 Explorative Datenanalyse 36
6.6.2 Inferentielle Statistik 37
6.6.2.1 Verbundenes und unverbundenes 2-Stichprobenproblem (abhängiger und unabhängiger t-Test) 37
6.6.2.2 Unverbundenes k-Stichprobenproblem (einfaktorielle unabhängige Varianzanalyse) 37
6.6.2.3 Faktorielle Pläne (mehrfaktorielle Varianzanalyse, ANOVA) 38
6.6.2.4 Faktorielle Pläne, multivariater Fall (Abhängige Varianzanalyse) 38
6.6.2.5 Nonparametrische Testverfahren 39
6.6.2.6 Bivariate Korrelation 40
6.6.2.7 Bivariate Korrelation höherer Ordnung (Partialkorrelation) 40
6.6.2.8 Multiple Regressionsanalyse 41
6.6.2.9 Faktorenanalyse 42
6.7 MAKRO-PROGRAMMIERUNG 42

1 ALLGEMEINE BEMERKUNGEN ZUR COMPUTERUNTERSTÜTZTEN DATENANALYSE

Die Verarbeitung der oftmals großen Datenmengen, die in einem psychologischen Experiment erhoben werden, erfolgt heutzutage nahezu ausschließlich mit Hilfe von Computern. Die Schnittstelle zwischen Mensch und Maschine stellen die sogenannten 'Sprachen' dar, mit welchen die Instruktionen zur Datenverarbeitung erstellt und dem Computer mitgeteilt werden. Die Form dieser Sprachen ist zu unterschiedlichen Graden der natürlichen Sprache angeglichen, so daß eine verständliche Art der Kommunikation zwischen Mensch und Maschine gewährleistet ist.

Es lassen sich grob 3 Arten von Sprachen unterscheiden:

1.1 SPRACHEN

1.1.1 Programmiersprachen

Diese sind sehr allgemein gehalten und lassen sich für verschiedenste Zwecke einsetzen. Solche Sprachen sind beispielsweise ADA, C, FORTRAN oder PASCAL. Gut geeignet für selbst erstellte Berechnungsprozeduren, womöglich unkonventioneller Art, verlangen diese jedoch eine gründlichere Kenntnis ihrer Syntax, die oft komplex und umfangreich sein kann. Darüber hinaus wird ein mehr oder weniger genaues Wissen über die interne Informationsrepräsentation und Informationsorganisation der jeweiligen Rechenanlage verlangt.

1.1.2 Betriebssystemsprachen

Diese Sprachen regeln den 'alltäglichen' Betrieb eines Rechners, wie zum Beispiel drucken oder speichern von Informationen, Dateiverwaltung etc. Die gebräuchlichsten Betriebsystemsprachen sind MS-DOS, OS/2 oder MS-Windows auf PCs, UNIX oder DEC/VMS auf Großrechnern. Die Betriebsystemsprachen sind eng verwoben mit den Programmiersprachen und Programmsystemen, besonders bei Großrechenanlagen. Aus diesem Grunde ist eine minimale Kenntnis der jeweiligen Betriebsystemsprache auch zur Anwendung von Statistik Programmpaketen wie SPSS erforderlich, weshalb auch eine kurze Einführung zum Betriebsystem DEC/VMS der Großrechenanlage des EDV-Zentrums der Universität Graz, auf welcher SPSS läuft, gegeben wird.

1.1.3 Programmsysteme

In solchen Systemen sind eine Vielzahl von sogenannten 'Einzweckprogrammen', wie sie sich mit Programmiersprachen erstellen lassen zusammengefaßt. Diese sind für eine bestimmte Aufgabe konzipiert, also zum Beispiel zur Datenverwaltung (z.B.: dBase) oder, wie es hier interessiert zur statistischen Datenerhebung und Auswertung. Die Befehlssprache ist relativ simpel, einfach zu erlernen und setzt kein umfangreiches Informatikwissen voraus. Bekannte statistischen 'Programmpakete' sind zum Beispiel SAS, P-STAT, CSS/STATISTICA, oder SPSS.

Letzteres zählt aufgrund seiner Vielseitigkeit und Verbreitetheit zu den beliebtesten statistischen Programmsystemen. Es läuft und bewährt sich seit mehr als 20 Jahren auf Großrechnern unter dem Namen SPSSX, auf PCs als SPSS-PC+ unter MS-DOS oder als SPSS für Windows (SPSSwin) unter der graphischen Benutzerumgebung MS-Windows und auch als Implementation für APPLE-Macintosh Maschinen.

1.2 BENUTZERINTERAKTION

Die Möglichkeiten, mit dem Programmsystem SPSS zu arbeiten, sind sehr vielseitig, zumal die Schwerpunkte der Handhabungsphilosophie zwischen den Plattformen, also Betriebsystemen, auf denen es installiert ist, differieren.

Es lassen sich 2 Grundprinzipien der Benutzer-SPSS Interaktion beschreiben:

1.2.1 Der Interaktive Betrieb

Hier erfolgen Dateneingabe, Befehlseingabe, Verarbeitung der Befehle und Ausgabe der Ergebnisse sukzessive. Das heißt, der Benutzer kann während jedes Berechnungsschrittes die Ergebnisse betrachten, gegebenenfalls die Rohdaten modifizieren andere Prozeduren ad hoc durchführen, schnell einige Grafiken erstellen usw. Obwohl die Möglichkeit des interaktiven Betriebs von SPSS auch am Großrechner gegeben ist, ist SPSS unter MS-Windows mit der Fenstertechnik und Menüsteuerung für diese Arbeitsweise prädestiniert. An dieser Stelle sollte jedoch erwähnt werden, daß das Programmpaket STATISTICA für Windows die erste Wahl ist, wenn man den interaktiven Betrieb bevorzugt.

1.2.2 Der 'Batch' Betrieb

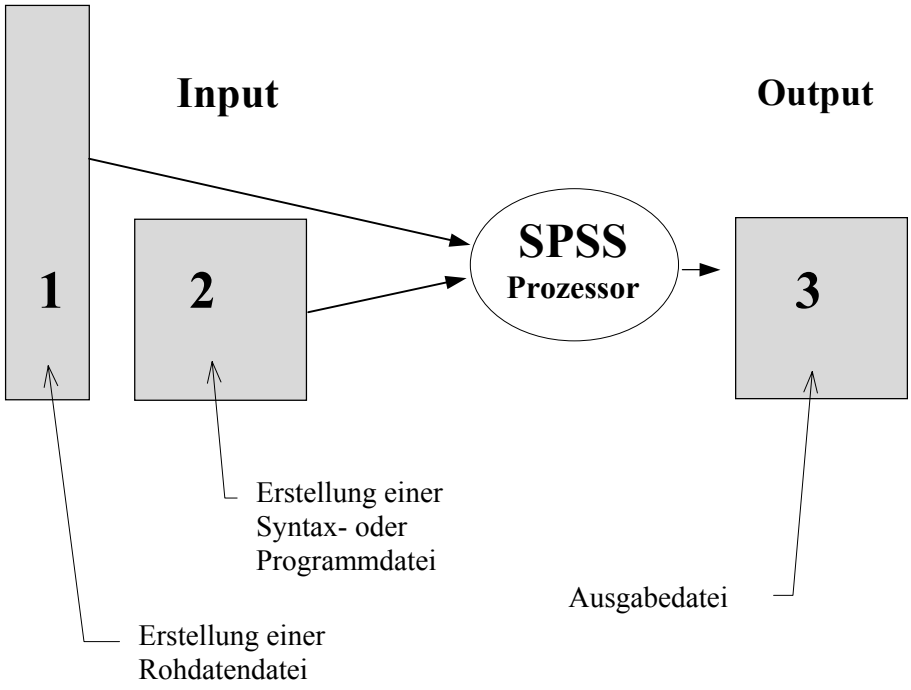
Diese Art der Kommunikation mit dem Rechner stellt die Haupt-Betriebsart von SPSS dar. Es werden Befehlsprogramme erstellt, das heißt, viele Befehle hintereinander in einer Datei, der sogenannten '*Batch-*' oder '*Stapeldatei*', zusammengefaßt. Diese werden vom Programmsystem SPSS bearbeitet und als Endprodukt der Auswertung wird eine Datei erstellt, in die sämtliche Berechnungsergebnisse geschrieben werden. Dies stellt sicherlich den 'wissenschaftlich saubereren' Einsatz des Systems dar, da die Prozeduren im Voraus geplant werden müssen und nicht, wie im interaktiven Betrieb möglich, während der Berechnung aufgrund von Zwischenergebnissen gewählt werden. Für diese Betriebsart sind alle SPSS Systeme, SPSSX, SPSS-PC+ und SPSSwin geeignet.

1.3 ☝ FAUSTREGELN

- Die Planung der Auswertung und die Wahl der Verfahren sollte im Vorhinein geschehen und nicht der Logik des Programmes folgen. Es könnte die Gefahr bestehen, nicht auf problemspezifische Fragestellungen einzugehen, sondern nur solche Prozeduren zu verwenden, die das System primär anbietet, die sich leicht verwirklichen lassen.
- Weiters sollte der Versuchung widerstanden werden, zu viele der umfangreichen Prozeduren zu verwenden, ohne zu prüfen, ob diese der Fragestellung überhaupt angemessen sind. Die Vielfalt an unterschiedlichen Ergebnissen führt zu Verwirrung und zu Interpretationen, die über die Fragestellung hinaus gehen.
- Die Kenntnis über die Benutzung eines solchen Programmpaketes wie SPSS kann nicht die mathematischen Hintergründe zu den Verfahren ersetzen, die zu einer sinnvollen Anwendung der Statistikprozeduren und eine korrekte Interpretation der Ergebnisse unerläßlich sind.

2 DAS GRUNDPRINZIP DES PROGRAMMSYSTEMS SPSS

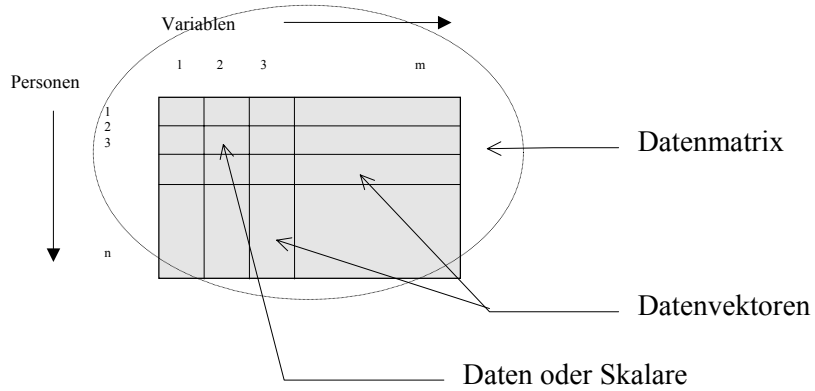
2.1 ALLGEMEINE STRUKTUR DER ARBEITSWEISE UND DIE 3 GRUNDLEGENDEN SCHRITTE



Schritt 1 = Dateneingabe: Die Rohdaten (z.B.: Ratings aus einem Fragebogen, Ergebnisse eines Experiments etc.) werden eingegeben. Diese Daten werden in der Rohdatendatei gespeichert.
Schritt 2 = Statistikprozeduren 'anfordern': SPSS-Befehle werden in eine weitere Datei, die sogenannte Syntax- oder Programmdatei geschrieben. Hier werden u.a. die durchzuführenden Statistikprozeduren definiert (z.B.: t-Test, Varianzanalyse, Korrelation etc.).
Schritt 3 = SPSS starten: Die beiden Dateien (Rohdatendatei und Syntaxdatei) werden von SPSS bearbeitet und es wird eine Ausgabedatei erstellt. In der Ausgabedatei oder dem Listing stehen die Ergebnisse der in der Syntaxdatei angeforderten Statistikprozeduren.

2.2 DATENSTRUKTUR 1

Die Daten, mit denen das SPSS System arbeitet, sind durchwegs Rohdaten, die so, wie sie beispielsweise in einem Experiment erhoben worden sind, ohne Umwandlung eingegeben werde. Sämtliche Berechnungen, Transformationen (z.B.: *Scorebildungen*) können vom SPSS System erledigt werden. Die Grundstruktur der Datenmatrix ist, daß **Variablen spaltenweise** und **Personen zeilenweise** eingegeben werden .



2.3 SYNTAXSTRUKTUR 2

Nach der Definition der Daten erfolgen die '*Anforderungen*' der Berechnungsprozeduren. Dies läuft immer nach dem gleichen Prinzip ab: ❶ Name der Prozedur, ❷ Liste der Variablen, ❸ Optionen.

2.4 SPSS AUF DER VAX

Unter VMS eignet sich SPSS besonders für den Batchbetrieb. Die Rohdaten und die SPSS Programme werden in eine oder mehrere Dateien geschrieben, anschließend wird das SPSS System gestartet und nach Berechnung eine Ausgabedatei mit den Berechnungsergebnissen angelegt. Diese Datei kann dann weiterverarbeitet werden.

Die SPSS Befehlssprache unterscheidet sich nur unwesentlich zwischen der Großrechner-, DOS- und Windows-Version, so daß auf Unterschiede im Zusammenhang mit den besprochenen Prozeduren eingegangen werden kann.

2.4.1 Vorteile

Bei großen Rohdatenumfängen (z.B.: EEG-Untersuchungen) und/oder Statistikprozeduren, die womöglich mit andere Programme (C, FORTRAN) interagieren, ist SPSSX auf dem Großrechner unbedingt vorzuziehen. Allgemein immer dann, wenn die Anzahl der zu untersuchenden Variablen größer als m=255 ist..

2.4.2 Nachteile

Die effiziente Handhabung setzt alleredings etwas Erfahrung mit SPSS und dem Betriebssystem VMS voraus. Für Anfänger und Auswertungen mit geringen Datenumfänge ist ein SPSS-System auf einem lokalen PC (SPSS PC+ bzw. SPSSwin) vorzuziehen.

2.5 SPSS UNTER WINDOWS

Unter MS-Windows auf PCs kann SPSS sowohl im interaktiven als auch im Batchbetrieb eingesetzt werden. Da die Arbeit mit Menüs und Fenstern selbsterklärend ist, und zum Verständnis der Logik der SPSS Befehlssprache nur abträglich wäre, soll hier nur der Batchbetrieb besprochen werden. Wenn man die Befehlssprache beherrscht und sich etwas unter Windows zurecht findet, so ist man auch in der Lage, SPSSwin interaktiv über Menüs zu betreiben. Zur Einführung in SPSSwin sei Kähler (1994) empfohlen.

Gestartet wird SPSS für Windows durch Anklicken des SPSS *Icons* in der SPSS Gruppe des Programmanagers (gilt für Windows 3.x). Die Befehle werden in das 'Syntax-Fenster' eingegeben und durch Anclicken der 'Ausführen' Taste bearbeitet. Die Ergebnisse werden dann sofort im 'Ausgabefenster' dargestellt. Dies kann auch mit mehreren Syntax und Ausgabefenstern erfolgen, die dann in weiterer Folge abgespeichert werden können.

2.5.1 Vorteile

Die Handhabung des Systems setzt keine fundierten Kenntnisse der SPSS-Syntax voraus. Die graphische Benutzeroberfläche erlaubt schnelles und effektives Arbeiten, besonders bei Erkundungsstudien und 'intuitiv' geführter Datenanalyse. In den meisten Fällen ist SPSSwin dem schwerfällig zu bedienendem SPSSX überlegen

2.5.2 Nachteile

Außer einer geringeren Datenkapazität sind kein Nachteile bekannt.

3 SPSS AUF DER VAX: SPSSX

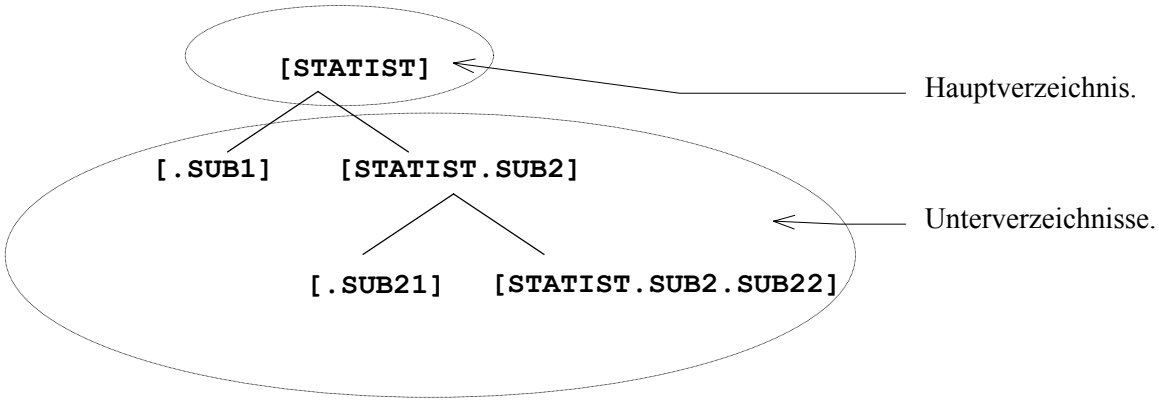
Da SPSS eng im Zusammenhang mit dem Betriebsystem des Rechners, auf dem es installiert ist, arbeitet, ist es notwendig, zu allererst die SPSS-Befehle von den Befehlen des Betriebsystemes abzugrenzen.

Der Großrechner an der Karl Franzens Universität Graz ist vom Typ VAX der Firma Digital Equipment Corporation, DEC mit dem Betriebssystem VMS oder kurz DEC/VMS. Hardwareinformationen können den Broschüren des EDV-Zentrums entnommen werden.

3.1 VMS VERZEICHNISSTRUKTUR

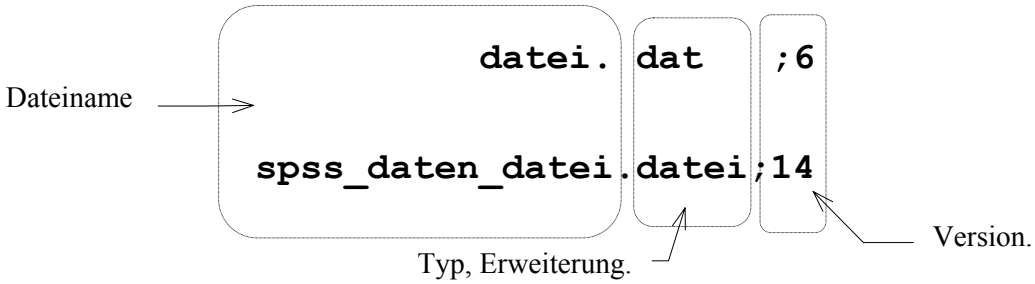
Die Hauptaufgabe von VMS bei der Arbeit mit dem Programmsystem SPSSX besteht darin, Dateien zu erstellen, zu bearbeiten und zu verwalten. Die Informationsorganisations-Struktur besteht aus sogenannten Verzeichnissen, in welchen *Dateien*, auch *Files* genannt oder andere Verzeichnisse enthalten sind. Man kann sich Verzeichnisse als 'Schubladen' oder 'Ordner' vorstellen, in welchen Informationen in Form von 'Schriftstücken' gelagert werden.

Verzeichnisse, auch *directories* genannt, werden in eckiger Klammer [] dargestellt, in der Hierarchie von links nach rechts durch Punkte getrennt, beispielsweise `[STATIST.SUB1.SUB11]`. Dabei bezeichnet der erste Name ganz links das Hauptverzeichnis oder *'Root directory'*, welches den *'Username'* trägt.



3.2 VMS DATEINOTATION

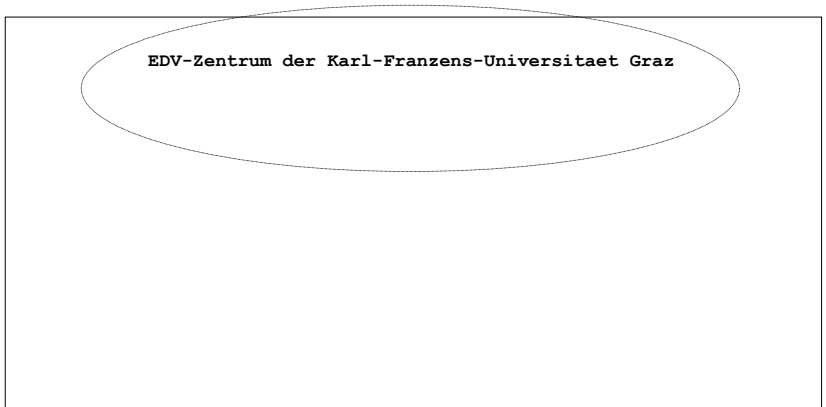
Dateien befinden sich in Verzeichnissen. Dateinamen, also die Namen von Informationsträgern, bestehen aus dem Namen, der bis zu 39 alphanumerische Zeichen erlaubt, einer Typbezeichnung oder Erweiterung, die durch einen Punkt getrennt hinter dem Dateinamen steht und der jeweiligen Versionsnummer der Datei, durch einen Strichpunkt getrennt, hinter der Typenbezeichnung. So bedeutet beispielsweise der Dateiname 'daten.dat;6' die 6 Version der Datei 'daten' vom Typ 'dat':

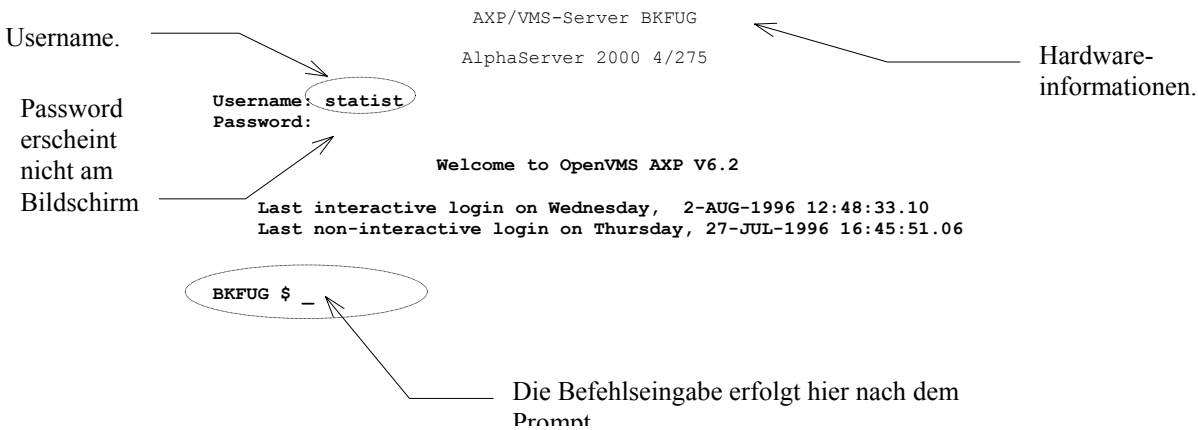


3.3 VMS STARTEN

Der Weg des *Einloggens* in den Großrechner, also des 'Startens', kann über verschiedene Wege erfolgen und ändert sich auch regelmäßig. Prinzipiell ist es notwendig, über einen *Account*, das bedeutet über eine Benutzerberechtigung, zu verfügen. Diese besteht aus einem *Username* und einem *Password*, welche bei jedem Einstieg in das System einzugeben sind.

Das VMS System arbeitet im allgemeinen analog zum bekannten MS-DOS System. Nach der Eingabe von *Username* und *Password* erscheint, nach informellen Meldungen, das $ *Prompt*zeichen, Befehle des Benutzers können nun eingegeben werden.





3.4 VMS BEFEHLE

Legende:

➔ **Befehl, Bef**	..... so heißt der Befehl und dessen Abkürzung.
Beispiel	..... das ist ein Beispiel zum Befehl.
<TASTE>	..... diese Taste muß gedrückt werden.

Befehle werden hinter dem *Prompt* (`BKFUG $`) eingegeben. Durch betätigen der Return-Taste (<RETURN>, <↵>, <ENTER>) werden die Befehle an das System übergeben, werden ausgeführt.

Es sei erwähnt, daß hier lediglich der notwendigsten Befehlssatz des überaus umfangreichen VMS Betriebssystems vorgestellt wird. Für tiefere Einblicke in das System sei auf die Kurzbeschreibung 'DEC/VMS' des EDV-Zentrums der UNI Graz verwiesen. Eine gute Zusammenfassung sämtlicher VMS-Befehle und eine profunde Beschreibung von Struktur und Arbeitsweise des Großrechner Systems VAX/VMS gibt Barthel (1990).

3.4.1 Verzeichnis erstellen

➔ **create/directory, create/dire**

Durch 'create/dire' wird ein neues Verzeichnis erstellt. Der gewünschte Name wird in eckiger Klammer dahinter geschrieben.

create/dire [.lade]

3.4.2 Verzeichnis wechseln

➔ **set default, set def**

Der Befehl ‘set def’, also set default, setzt das aktuelle Verzeichnis, in dem gearbeitet wird. Dabei folgt dem Befehl ‘set def’ der Verzeichnisname entweder als vollständige *Pfad*, z.B.: `[statist.sub1]` oder lediglich als Name des Verzeichnisses, das in der Hierarchie eine Stufe unterhalb liegt, z.B.: `[.sub1]`.

```
set def [.sub1]
set def [-]
```

‘Set def’, gefolgt von einem Bindestrich in eckiger Klammer wechselt in ein Verzeichnis, das eine Stufe höher in der Verzeichnishierarchie steht. ACHTUNG: Dieser Befehl wird auch dann ausgeführt, wenn gar kein höheres Verzeichnis besteht. Man befindet sich dann in einem nicht existenten Verzeichnis, z.B. `[00000.00000.00000]`. Durch `set def [root]` gelangt man in sein Hauptverzeichnis zurück.

3.4.3 Editor-Befehle

Editoren stellen die beste Möglichkeit dar, neue Dateien zu erstellen oder vorhandene zu bearbeiten.

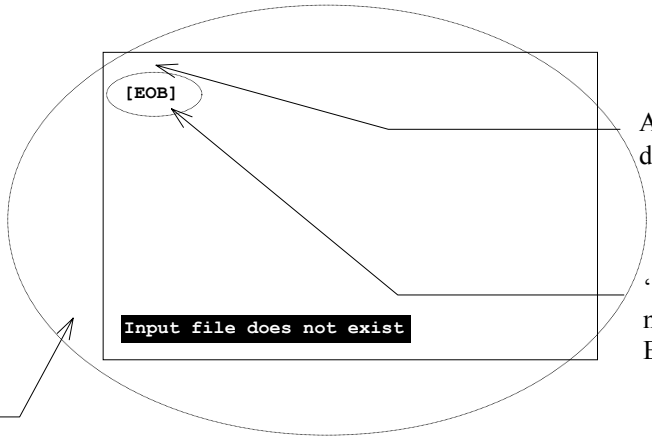
3.4.3.1 Dateien erzeugen

➔ **`edit`**

‘Edit’ ruft einen *Editor* des VMS Sytems auf, mit welchem die Informationen eingegeben, bearbeitet und in Form von Dateien gespeichert werden können.

```
edit daten.dat
```

Der Bildschirm sieht dann folgendermaßen aus:



Ab hier erfolgen die Eingaben.

‘**E**nd **O**f **B**uffer’ markiert das Ende der Datei.

Editor-Oberfläche.

Die ‘Navigation’ im Editor erfolgt über die Cursor Tasten der Tastatur. Das Markieren und Löschen bestimmter Textteile variiert von Terminal zu Terminal. Beste Methode:Trial and Error.

3.4.3.2 *Editor beenden, Dateien speichern*

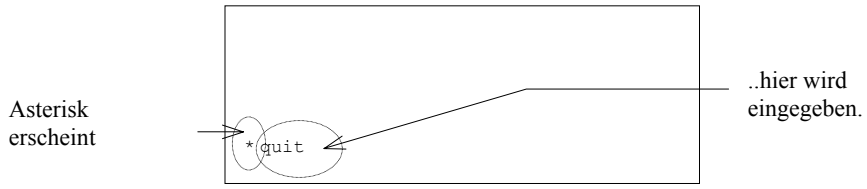
<STRG> z oder **<CTRL> z**

Durch gleichzeitiges Betätigen der Steuertaste <STRG> oder auch <CTRL> und der 'Z' Taste beendigt man die Editorsitzung, es erscheint ein *Asterisk* *****. Nach diesem Zeichen werden entweder `quit` oder `exit` eingegeben:

Editor beenden ohne Abspeichern:

➔ **quit**

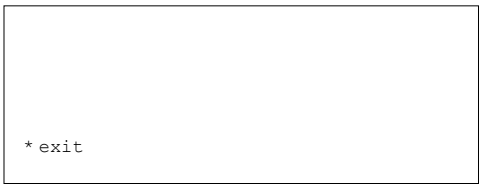
Die Eingabe 'quit' beendigt den Editor, ohne die Änderungen an der Datei zu speichern, es wird keine neue Datei, bzw. Dateiversion angelegt.



Editor beenden mit Abspeichern:

➔ **exit**

Die Eingabe 'exit' beendigt den Editor und speichert die Änderungen ab, eine neue Datei oder Dateiversion wird angelegt.



Man kehrt wiederum zur VMS Befehlsebene zurück, das Prompt erscheint.

3.4.4 Dateien löschen

➔ **delete, del; purge, pur**

del daten.dat;2

delete löscht eine bestimmte Datei. Dabei muß ‘del’, gefolgt von dem Dateinamen der zu löschenden Datei, mit Versionsnummer angegeben werden. Da Verzeichnisse ebenfalls als Dateien gelten, können diese ebenfalls mit ‘del’ gelöscht werden, vorausgesetzt, daß diese keine Dateien enthalten,

purge

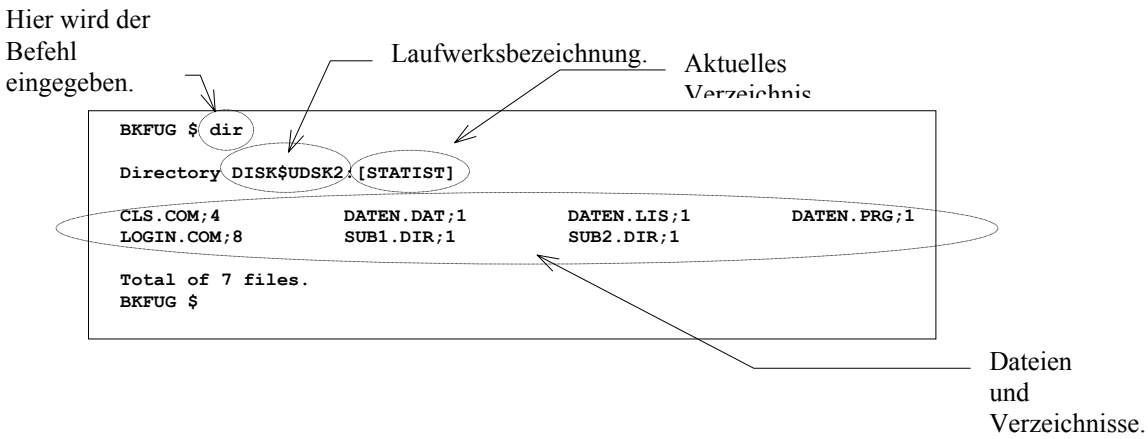
löscht die alten Versionen sämtlicher Dateien im aktuellen Verzeichnis. Es bleiben jene Dateien mit der höchsten, also aktuellsten Versionsnummer erhalten.

3.4.5 Verzeichnis und Dateien auflisten

➔ **directory, dir**

Der Befehl ‘dir’ zeigt an, in welchem Verzeichnis man sich befindet und listet sämtliche enthaltenen Dateien auf:

dir



3.4.6 Aussteigen

➔ **logout, lo**

Mit ‘lo’, logout beendigt man die Arbeit am Großrechner und kehrt zur lokalen Ebene zurück, zu MS-DOS oder MS-Windows . Dies soll immer nach jeder VMS Sitzung erfolgen

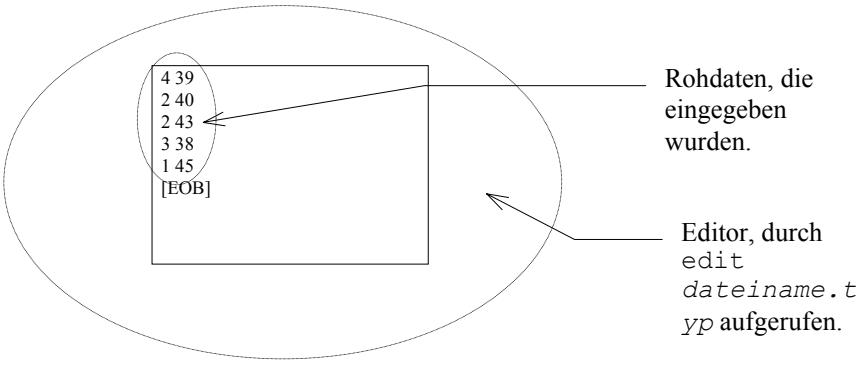
lo

3.5 SPSSX

3.5.1 Dateneingabe

Die Datenmatrix wird über den Editor eingegeben. Die Rohdatendatei erhält allgemein die Erweiterung ‘dat’, sofern diese im ASCII - Format erstellt ist.

Es wurden beispielsweise an 5 Studenten die Maturanote in Mathematik und die Schuhgröße erhoben. Die Datenmatrix sieht, wenn man die Mathematiknote in die erste, die Schuhgröße in die dritte und vierte Spalte schreibt, folgendermaßen aus:

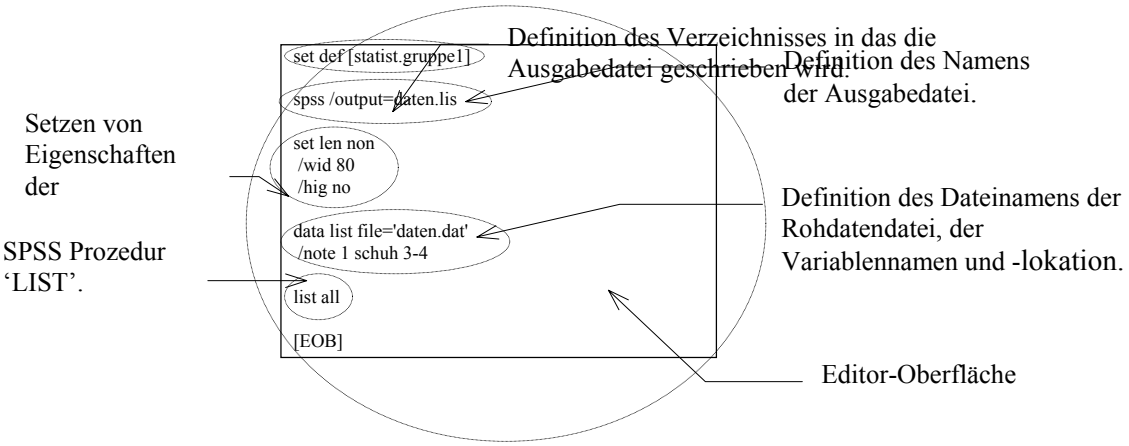


Ob man ein oder mehr Leerzeichen zwischen die Variablen setzt oder nicht, ist unerheblich, da die Lokation der Variablen erst in der Programmdatei genau definiert wird (siehe unten). Der besseren Übersicht halber sollte allerdings mindestens ein Leerzeichen zwischen die Daten geschrieben werden. Dadurch wird auch die Weiterverarbeitung mit anderen Programmsystemen wie zum Beispiel Excel, Word, o.ä. und Programmiersprachen -meist FORTRAN- vereinfacht.

Der Ausstieg aus dem Editor und die Speicherung der Rohdatendatei erfolgt mit `<STRG> z`, speichern mit `exit.`

3.5.2 Syntaxeingabe

Die SPSS- Syntax mit den Prozedurbefehlen wird ebenfalls wie die Rohdaten über den Editor eingegeben. Syntax- oder Programmdateien erhalten die Erweiterung ‘prg’. Grundprinzip ist, daß (a) zuerst die Variablen der Rohdaten bzw. deren Position definiert werden. Hier ist es möglich die Rohdaten direkt in die Programmdatei zu schreiben, was bei kleineren Datenmengen vor allem im interaktiven Betrieb bei SPSSX vorteilhaft ist. Werden die Rohdaten in eine eigene Datei geschrieben, dann wird vorher der Name der zu verwendenden Datendatei angegeben. Nach der Definition der Daten erfolgen (b) die ‘Anforderungen’ der Berechnungsprozeduren. Dies läuft immer nach dem gleichen Prinzip ab: Name der Prozedur, Liste der Variablen, Optionen.



Der Ausstieg aus dem Editor und die Speicherung der Syntaxdatei erfolgt wie in 3.4.3.2 beschrieben: Aussteigen mit `<STRG> z`, speichern mit `exit.`

3.5.3 Berechnung

➔ **spssx**

Die Berechnung erfolgt durch den Befehl 'SPSSX', gefolgt vom jeweiligen Programmdateinamen. Dieser Befehl wird auf der VMS-Befehlsebene (nach dem Prompt BKFUG $) eingegeben.

```
spssx daten.prg
```

SPSS greift bei der Ausführung auf die Datei 'daten.prg' zu, die Definition von Rohdaten bzw. Rohdatendatei, Ausgabedatei und Ausgabeverzeichnis erfolgt am Beginn des SPSS Befehlsprogrammes. Am Terminal erscheint folgende (oder eine ähnliche) Meldung:

```
Job DATEN (queue FAST$BATCH1, entry 44)started on FAST$BATCH1
```

Die Befehle der Syntaxdatei werden bearbeitet. Nach kurzer Zeit erscheint die Nachricht, daß die Berechnungen durchgeführt wurden, die Ausgabedatei wurde im definierten Verzeichnis abgelegt:

```
Job DATEN (queue FAST$BATCH1, entry 44) completed
```

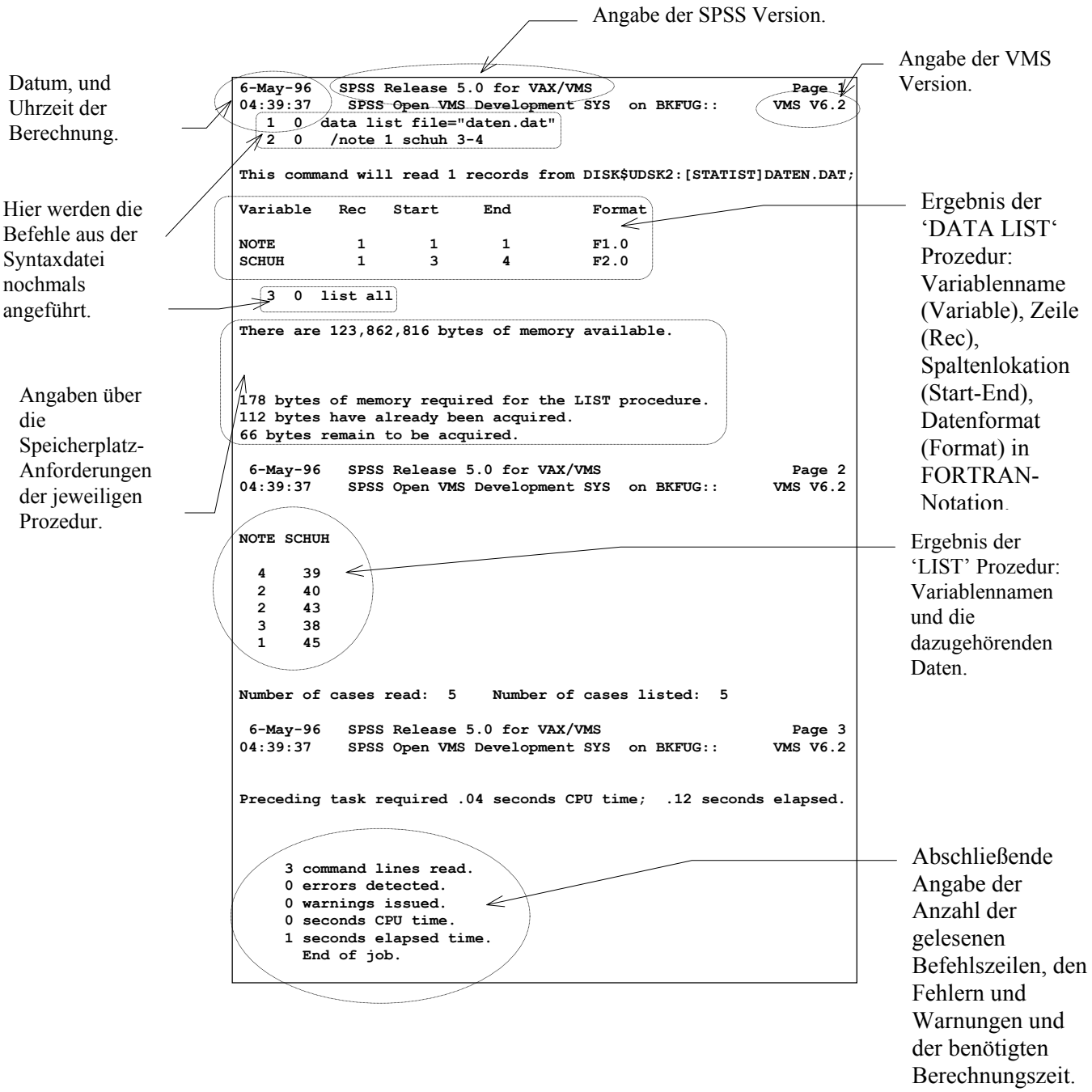
Ist diese Nachricht erschienen, so können die Berechnungsergebnisse in der Ausgabedatei betrachtet werden.

Dies ist eine Möglichkeit der SPSS Ausführung im Batchbetrieb, auf weitere Möglichkeiten wird hier nicht eingegangen.

3.5.4 Ergebnis

Die Ausgabedatei wird mit dem Editor aufgerufen. Diese Ausgabe-Dateien enthalten das Ergebnis der Berechnung und haben die Erweiterung ‘lis’. Die Bezeichnungen sind nicht obligat, sollten jedoch der allgemeinen Verständlichkeit halber beibehalten werden.

Eine SPSS Ausgabedatei hat folgenden Aufbau:

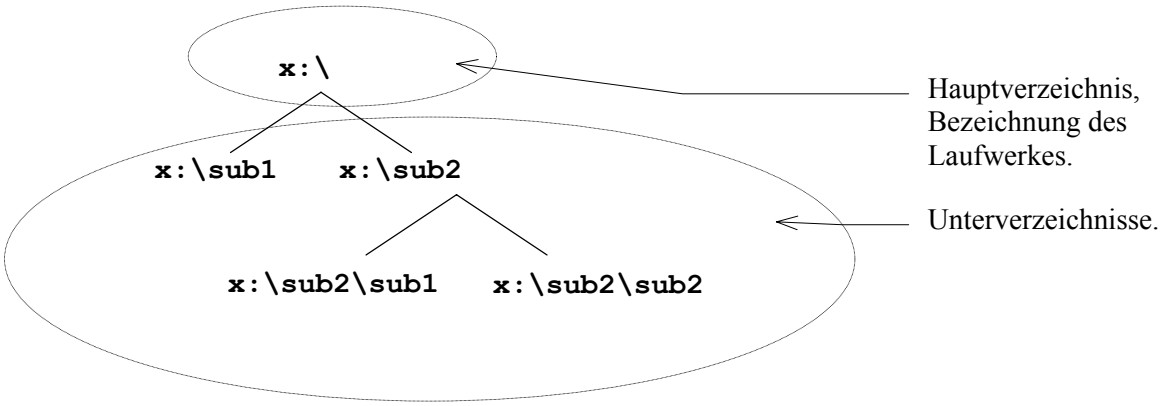


3.5.5 Weitere Möglichkeiten

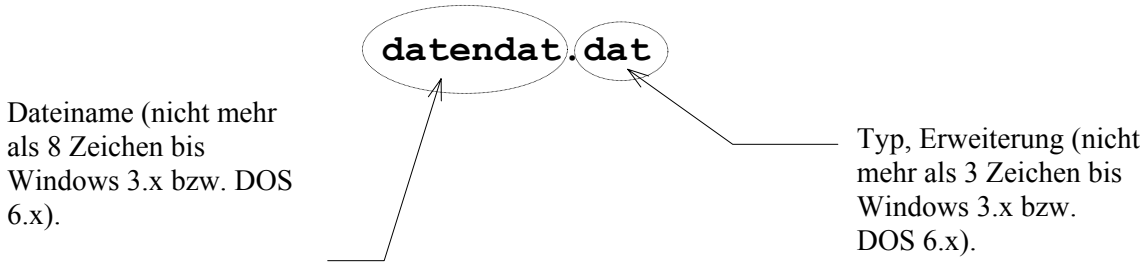
Die Ausgabedatei kann nun weiterverarbeitet werden. Um die Datei auszudrucken und/ oder in andere Programme (z.B.: Word) einzufügen, ist es notwendig, die Ausgabedatei auf den lokale PC zu übertragen. Für diesen Transfer vom Großrechner zum PC stehen verschiedene Protokolle, Programme zur Verfügung (FTP, Kermit, Z-Modem etc.). Der Umgang mit diesen Programmen kann den, im EDV-Zentrum aufliegenden Broschuren, entnommen werden. Eine ganz einfache Vorgangsweise ist das Kopieren (vom Großrechner) und das Einfügen der gewünschten Textdatei (in eine Anwendung am PC).

4 SPSS UNTER WINDOWS: SPSSWIN

4.1 VERZEICHNISSTRUKTUR



4.2 DATEINOTATION

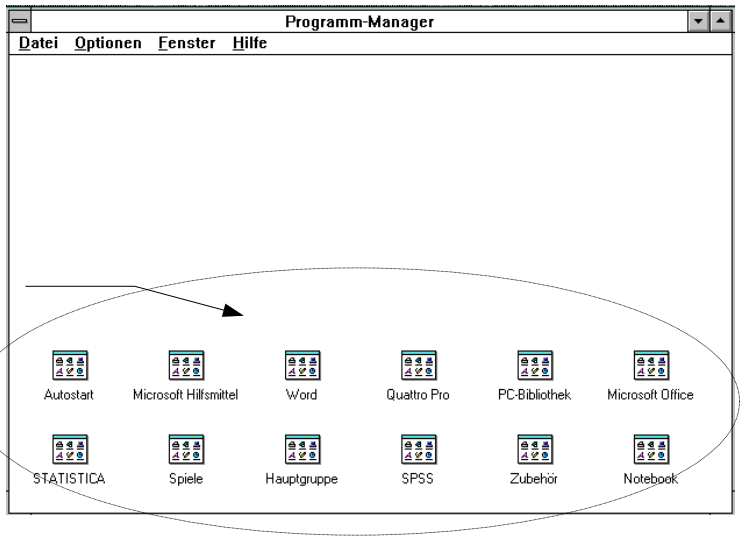


4.3 STARTEN

Wall (Lehrveranstaltung): Username und Password eingeben. Diese werden vom Lehrveranstaltungsleiter bekanntgegeben. Man gelangt direkt zur Windows Oberfläche.
Sonst: Windows je nach der Rechnerkonfiguration starten, meist durch die Eingabe von `win`.

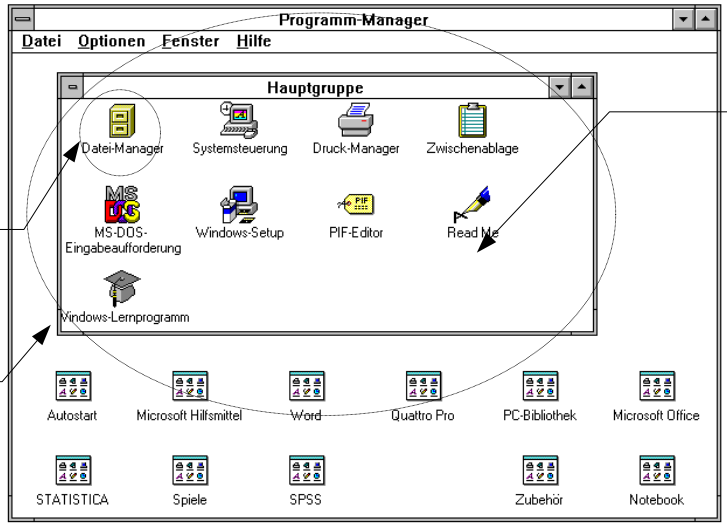
4.4 GRUNDLEGENDE 'MOUSECLICKS'

Das sind Programmgruppen Icons. Sie symbolisieren eine Gruppe zusammengehörender Programme



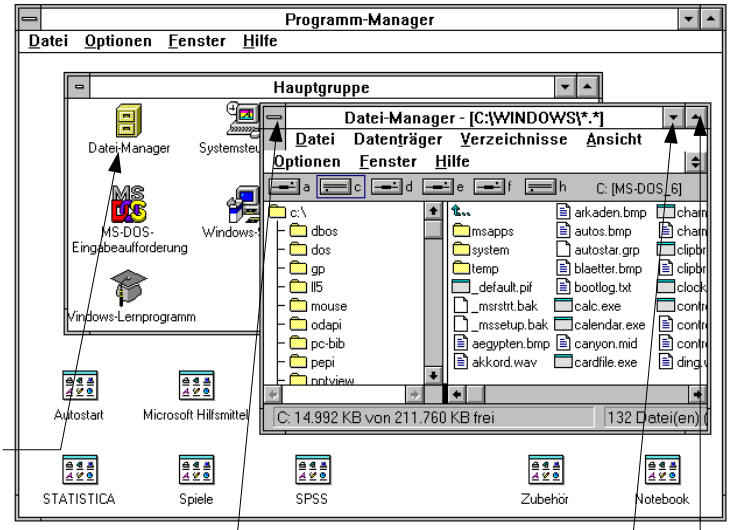
Icon oder Symbol eines Programmes.

Das ist ein Fenster.



Durch Doppelclicken auf eines der Programmgruppen-Icons wird ein Fenster geöffnet in dem wiederum neue Icons erscheinen. Diese Icons symbolisieren nun einzelne Programme. Hier am Beispiel der Programmgruppe 'Hauptgruppe' demonstriert.

Doppelclickt man nun auf eines der Programm-Icons, so wird ein Programm gestartet. Hier demonstriert am Programm ‘Datei-Manager’. (Mit diesem Programm könne Verzeichnisse erstellt, Dateien geöffnet verschoben,



Beenden des Programmes durch Doppelclicken.

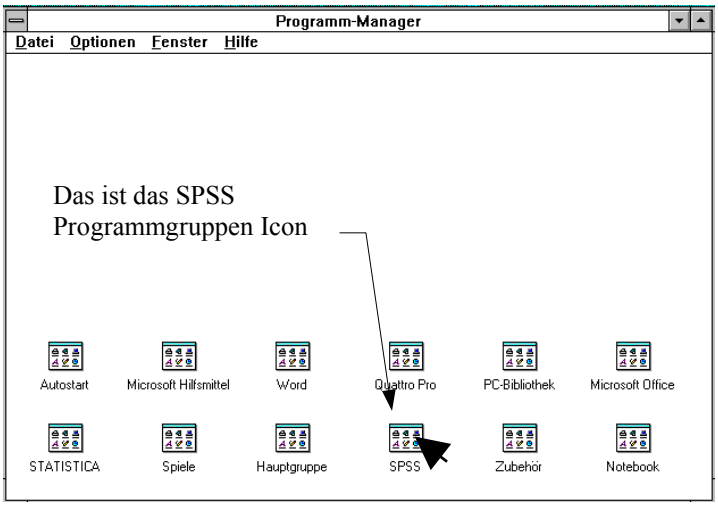
Wird hier angeclickt, so wird aus dem Fenster ein Icon.

Wird hier angeclickt, so füllt das Fenster den gesamten Bildschirm aus.

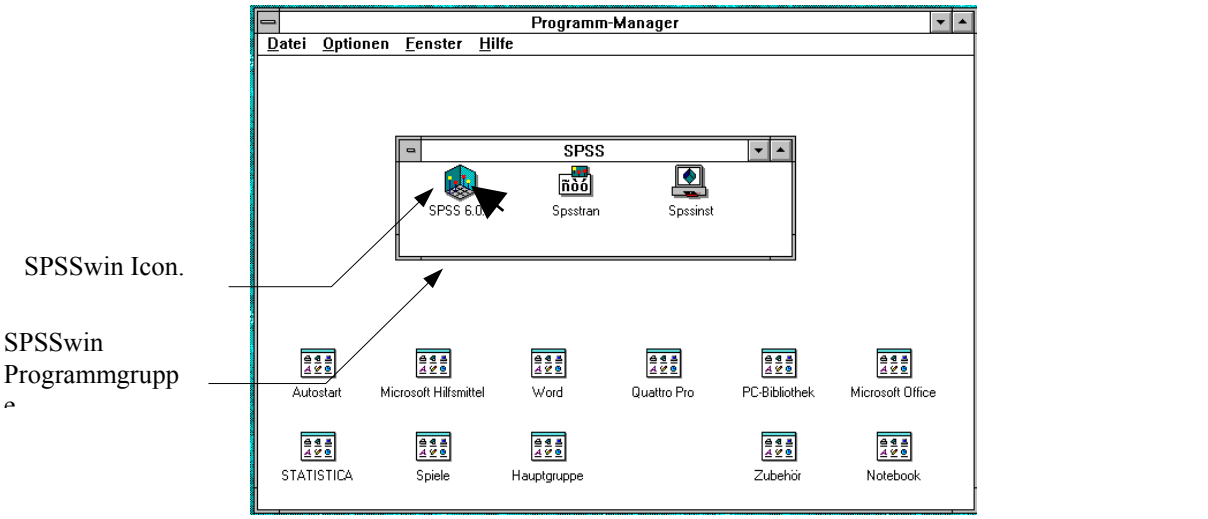
4.5 SPSSWIN

4.5.1 Starten

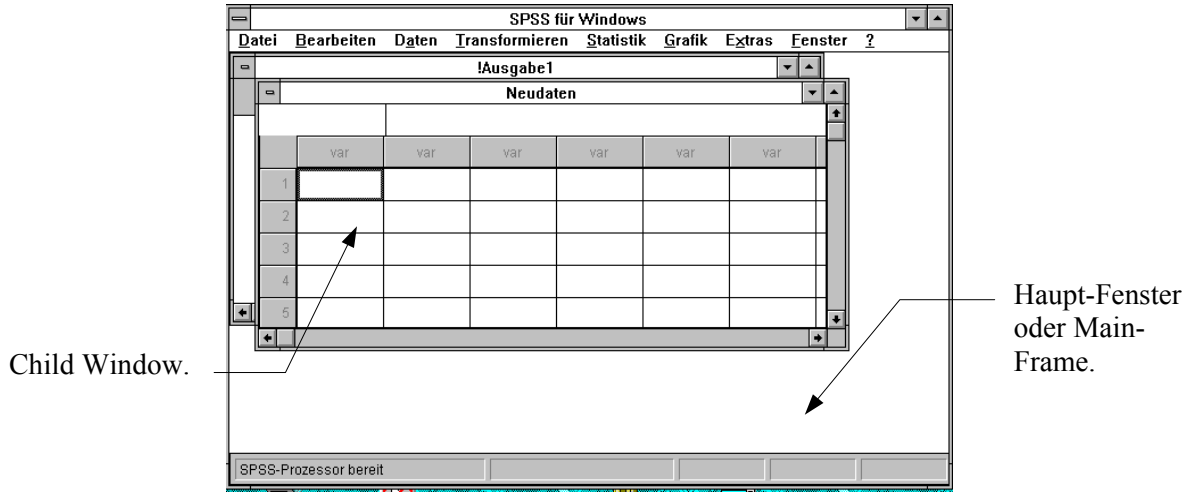
Zuerst wird die SPSS-Programmgruppe durch Doppelclicken geöffnet (s.u.).



SPSS wird gestartet indem auf das SPSS Programm-Icon in der SPSS Programmgruppe doppelt geclickt wird.

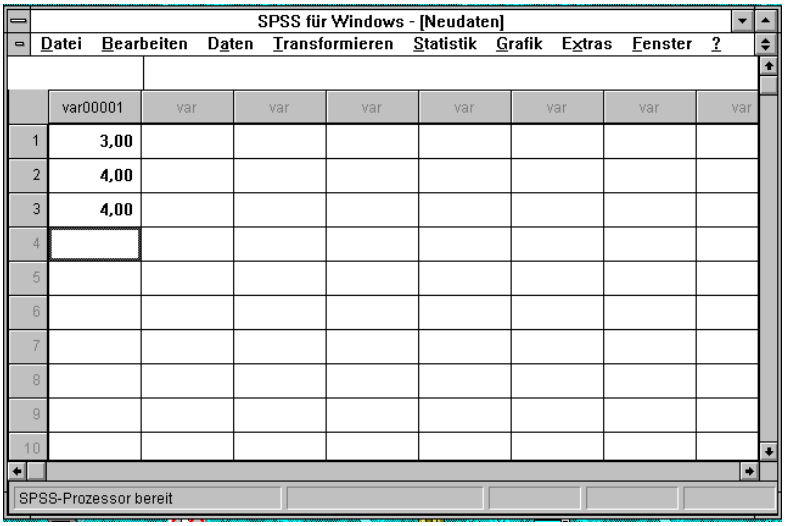


Es erscheint die SPSSwin Oberfläche. Diese Besteht aus einem Hauptfenster (Main-Frame) und untergeordneten Fenstern (Child-Windows), in welche die Rohdaten und die SPSS-Befehle eingegeben werden können. Die SPSS-Ausgabe der Ergebnisse erfolgt in einem weiteren Fenster, dem Ausgabefenster.

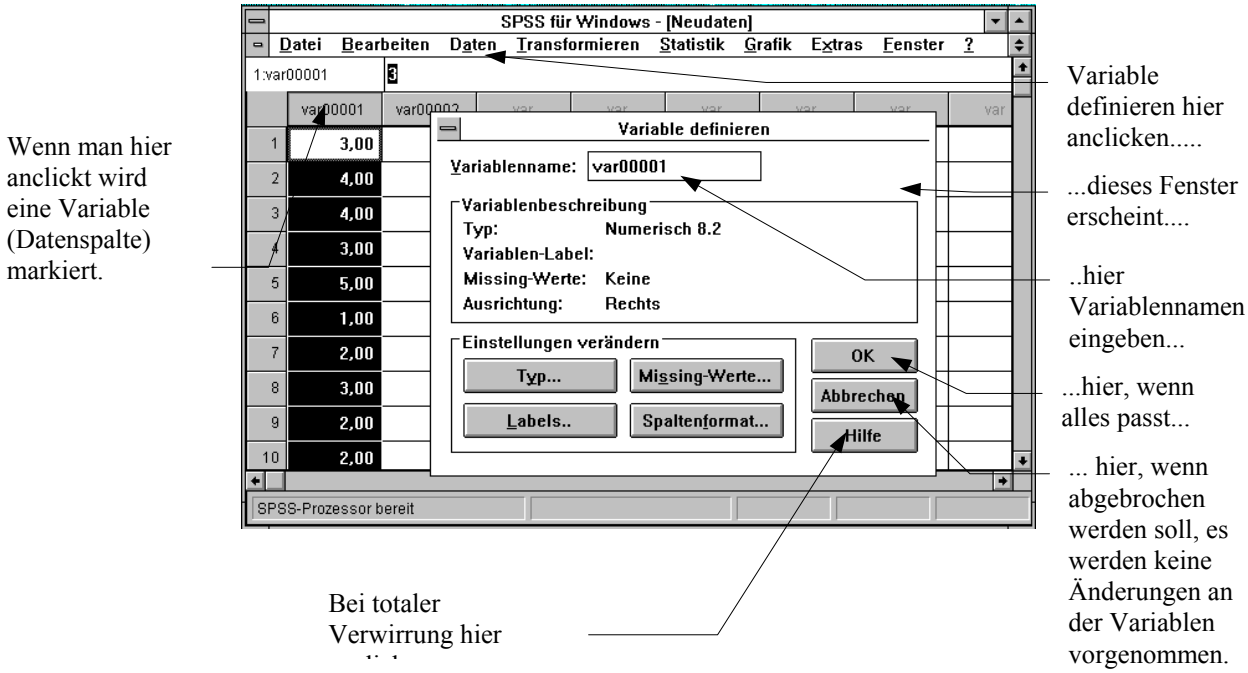


4.5.2 Dateneingabe

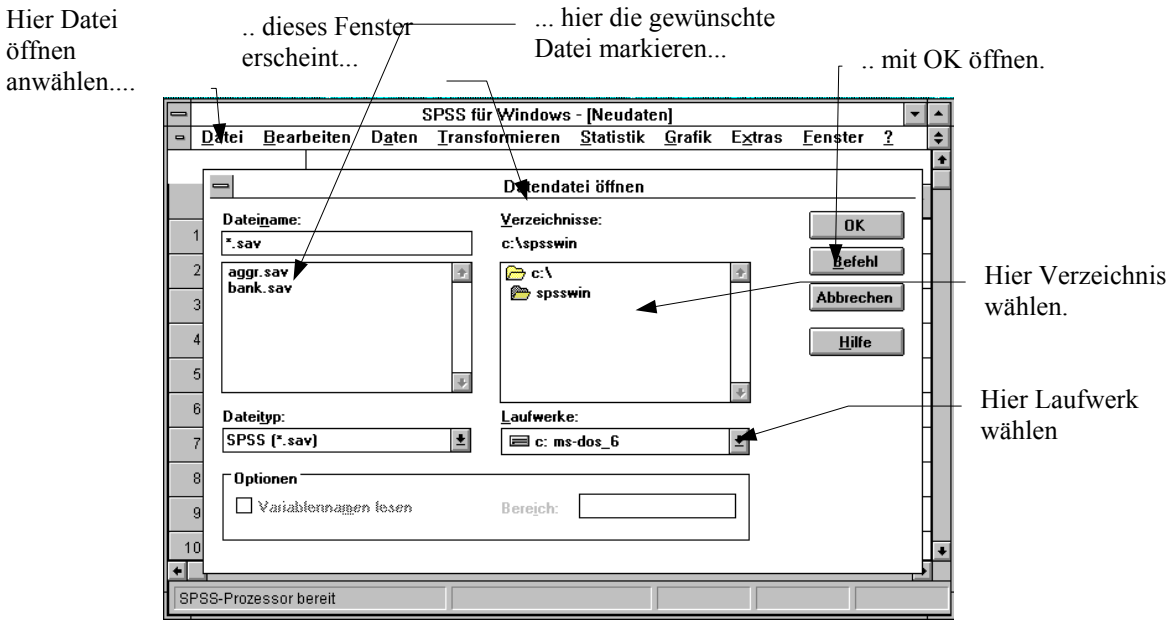
Die Rohdaten werden in das Fenster [Neudaten] eingegeben.



Durch anclicken des Menüpunktes Daten.Variable definieren werden die Eigenschaften der vorher markierten Variablen (Name, Format etc.) festgelegt.

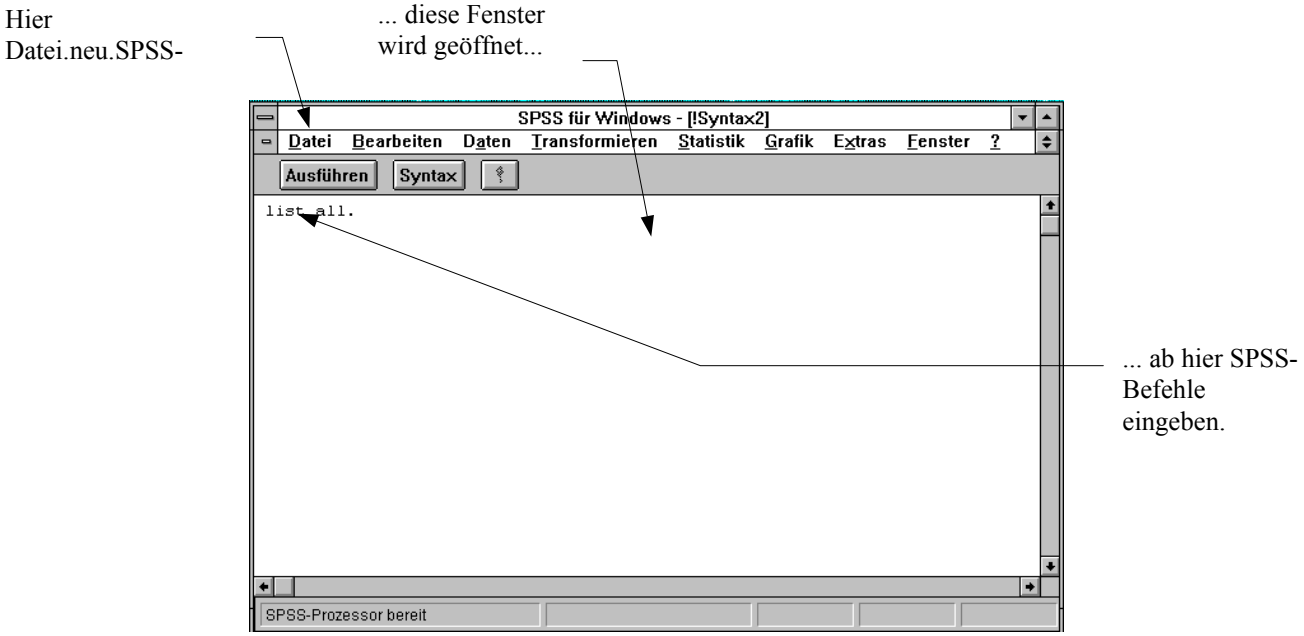


Durch Anwahl des Menüpunktes Datei.öffnen wird eine gespeicherte Rohdatendatei geöffnet.

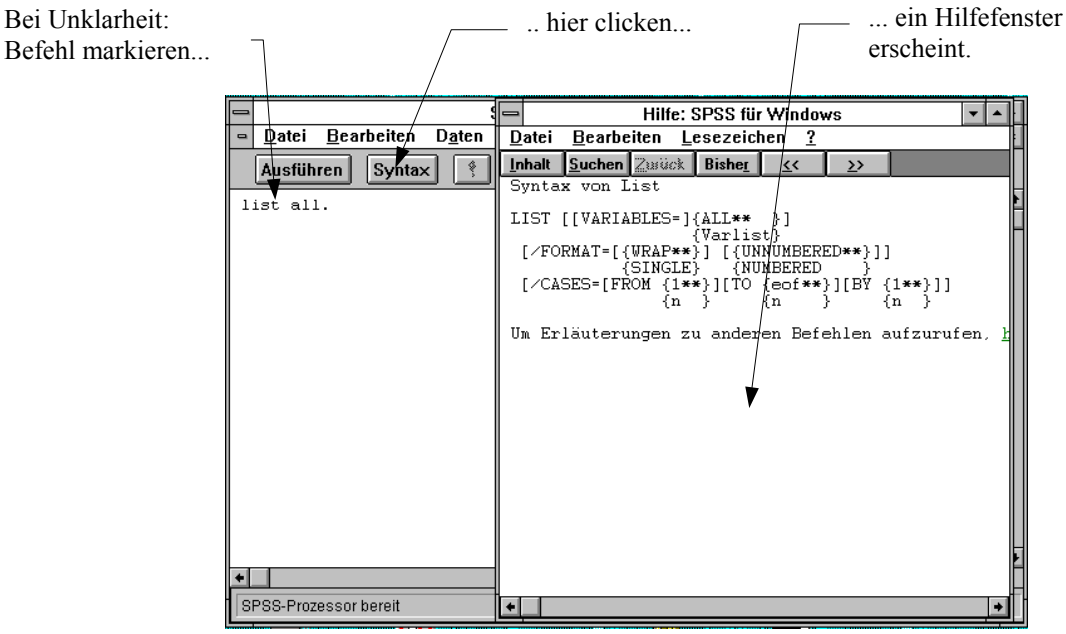


4.5.3 Syntaxeingabe

Falls kein Syntax Child window vorhanden ist, wird dieses durch Datei.neu.SPSS-Syntax geöffnet.

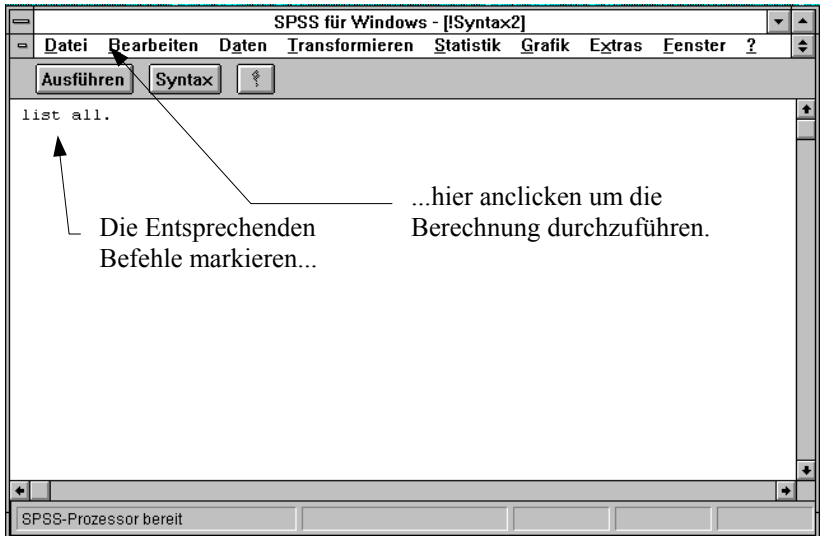


Die SPSS Befehle werden in das Fenster [Syntax] eingegeben. Will man die Syntax-Struktur eines SPSS-Befehls wissen, so wird der jeweilige Befehl markiert und auf den Knopf Syntax geclickt. Ein Hilfefenster erscheint.



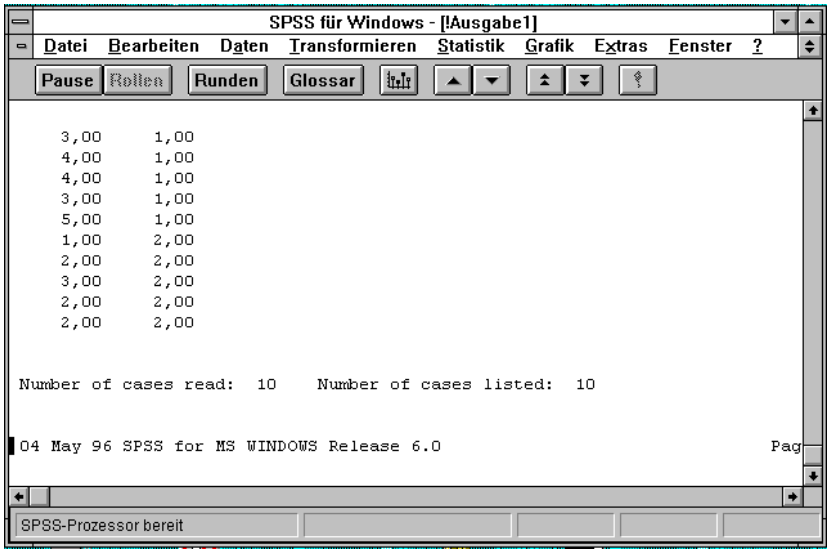
4.5.4 Berechnung

Um einen SPSS-Befehl auszuführen wird die entsprechende Stelle markiert und der Knopf Ausführen gedrückt.

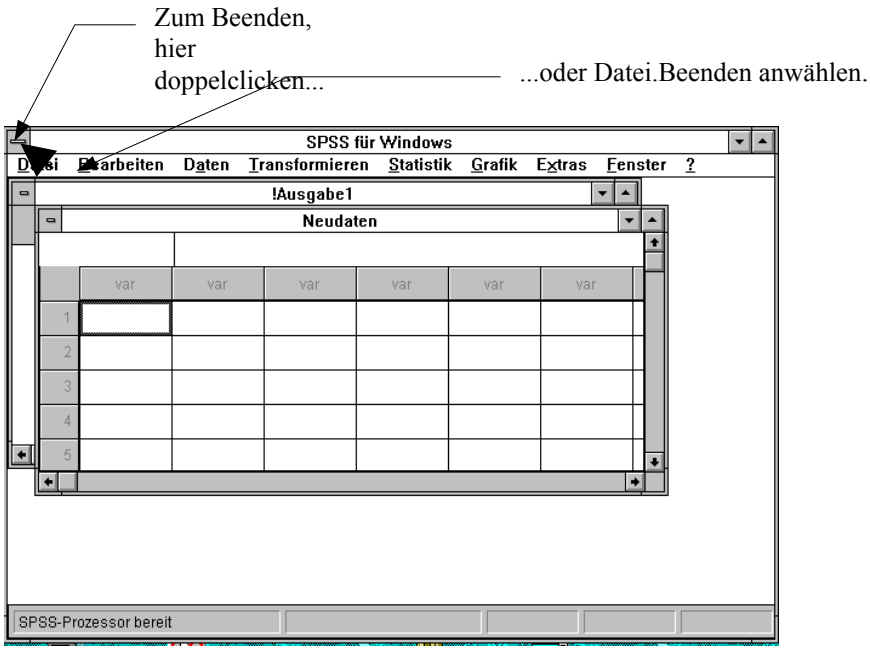


4.5.5 Ergebnis

Die Ergebnisse der Berechnung erscheinen im Ausgabefenster. Diese können durch den Menüpunkt Datei.Ausgabe Speichern unter einem bestimmten Namen abgespeichert werden.

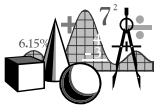


4.5.6 Aussteigen



5 BEISPIELE

Rechenbeispiele werden in der Lehrveranstaltung gebracht und können im World Wide Web unter der URL **http://bkfug.kfunigraz.ac.at/~statist/stats.htm** (weltweit und zu jeder Zeit) nachgelesen werden.



6 SPSS-BEFEHLE UND PROZEDUREN

6.1 ALLGEMEIN

Die Struktur der SPSS-Befehle ist im allgemeinen folgendermaßen:

Prozedur Variablenliste /Option
/Option /Option
/Option.

Legende:

➔ **Befehl, Bef** .. so heißt der Befehl und dessen Abkürzung.

Beispiel ..... das ist ein Beispiel zum Befehl.

Syntax ..... so ist die Befehlsstruktur (Eingabekonvention, Optionen etc.):

. Das in SPSSwin notwendige Zeichen zum Befehlsabschluß, der Punkt, erscheint nicht im Syntaxdiagramm.

A Elemente in Großbuchstaben sind Schlüsselwörter. Sie müssen wie angegeben eingegeben werden, können jedoch abgekürzt werden, solange dadurch kein Mißverständnis entsteht.

a.Elemente in Kleinbuchstaben sind benutzerspezifische Angaben.

[ ] Elemente in eckigen Klammern sind wahlfrei.

( ), ‘, „ Runde Klammern, einfache Anführungszeichen und doppelte Anführungszeichen sind unbedingt erforderlich, wenn sie angegeben sind, außer wenn sie in eckigen Klammern stehen.

‘ = „ Ein Paar einfacher Anführungszeichen ist in der Regel einem Paar doppelter Anführungszeichen gleichgestellt.

{ } Geschweifte Klammern zeigen eine Auswahlmöglichkeit zwischen verschiedenen Elementen an.

varlist Das Wort Varlist steht für eine Liste von Variablennamen.

6.2 ROHDATEN EINLESEN

6.2.1 ASCII-Daten einlesen:

➔ **data list, data lis**

Sofern mit Rohdaten im ASCII Format gearbeitet wird, definiert ‘data list’ die Rohdatenstruktur, also in welcher Spalte, welches Format, welcher Name und , falls die Rohdaten in einer eigenen Date stehen, den Dateinamen.

data list file=‘rohdat.dat’ notable /sex 1 age 3-4

```
DATA LIST [FILE=Datei] [{FIXED}] [RECORDS={1}] [{TABLE   }]
                       {FREE }           {n}   {NOTABLE}
                       {LIST }
/{1    } Varlist {Spaltenpos [(Format)]   } [Varlist ..]
[/{2    } ...] [/ ...]
```

6.2.2 Binär-Daten einlesen:

➔ **get file, get fil**

Sind die Rohdaten im binär Format vorhanden, häufig bei SPSSwin, so definiert man die Rohdatendatei mit ‘get file’. Spezifikationen der Variablen sind nicht nötig, da diese in der binären Datendatei erstellt wurden.

get file eq ‘daten.sav’

```
GET FILE=Datei
 [/KEEP={ALL    }] [/DROP=Varlist]
        {Varlist}
 [/RENAME=(alte Varlist=neue Varlist)...]
```

6.2.3 Direkte Dateneingabe:

➔ **data list, data lis**

Sollen Rohdaten und SPSS-Befehle in einer Datei stehen, dann schreibt man die Datensätze zwischen die Anweisungen ‘begin data ‘ und ‘end data’ in die SPSS Syntax-Datei..

```
data list notable /age 1-2
begin data
32
32
12
24
43
end data
```

```
BEGIN DATA
Datensätze
END DATA
```

6.3 ROHDATEN SPEICHERN

6.3.1 Als ASCII-Datei speichern:

➔ **write, wri**

```
wri out='x:\dat\n_daten.dat' /all.
```

Alle zur Zeit im Hauptspeicher befindlichen Daten werden im ASCII Format unter dem Dateinamen n_daten.dat abgespeichert. Die so gespeicherten Daten können mit der data list Prozedur wiederum eingelesen werden.

```
WRITE [OUTFILE=file] [RECORDS={1}]
                              {n}
   [{NOTABLE}]
    {TABLE  }
 /{1    } varlist [{col location [(fmt)]}]
  {rec #}          {(fmt list)          }
                   {*                   }
         [varlist...]
 /{2    }...]
  {rec #...}
```

6.3.2 Als Binär-Datei speichern:

➔ **save, sav**

```
sav out="x:\dat\bsp_11.sav" /com.
```

Alle zur Zeit im Hauptspeicher befindlichen Daten werden im Binär Format unter dem Dateinamen bsp_11.sav in komprimierter Form abgespeichert. Die so gespeicherten Daten können mit der 'get file' Prozedur wiederum eingelesen werden.

```
SAVE OUTFILE=file
 [/KEEP={ALL    }] [/DROP=varlist]
        {varlist}
 [/RENAME=(old varlist=new varlist)...]
 [/MAP] [/{COMPRESSED  }]
          {UNCOMPRESSED}
```

6.4 ROHDATENMANIPULATION

6.4.1 Berechnung neuer Variablen:

➔ **compute, comp**

Oft ist es von Nöten, neue Variablen, meist Scores aus Rohdaten zu berechnen. Mit 'compute' wird eine neue Variable nach der angegebenen Berechnungsvorschrift erstellt oder eine alte modifiziert, die sogenannte Zielvariable.

```
compute (score) eq var1 + var2 + var3
```

dies kann auch durch
compute (score) eq sum(var1 to var3)
ausgedrückt werden, was bei längeren Reihen von gleichlautenden Variablen mit Laufindex nützlich ist. Deshalb sollte schon in der Rohdatendefinition (data list) eine entsprechende Variablenbezeichnung gewählt werden.

```
COMPUTE  Zielvariable = Ausdruck

Arithmetische Operatoren:
+     Addition          -     Subtraktion
*     Multiplikation    /     Division
**    Potenzierung

Statistische Funktionen:
SUM[.n]           Argumentesumme
MEAN[.n]          Mittelwert der Argumente
SD[.n]            Standardabweichung der Argumente
VARIANCE[.n]      Varianz der Argumente
MIN[.n]           Argumente-Minimum
MAX[.n]           Argumente-Maximum
```

6.4.2 Bedingte Berechnung neuer Variablen:

➔ **if**

Soll nur unter bestimmten Bedingungen ein Wert einer Variablen berechnet werden, so ist ‘if’ zu verwenden. Der Zuordnungsteil der Befehlssyntax von ‘if’ entspricht der Syntax des Befehls ‘compute’.

If (sex eq 1 and age lt 30) neuvar eq 2
die Variable ‘neuvar’ soll den Wert 2 erhalten, falls die Person männlich ist und jünger als 30 Jahre.

```
IF [(]logischer Ausdruck[)] Zielvariable=Ausdruck
```

6.4.3 Schleifenbildung:

➔ **loop, loo; end loop, end loo**

Bestimmte Prozeduren werden wiederholt durchgeführt. Die Prozeduranweisung steht zwischen ‘loop’ und ‘end loop’.

```
loop i=1 to 1000.
+ compute iq = (normal(15)+100).
write out="dmy.dat" /iq.
end loop.
```

Hier wird ein Wert der Variable iq nach einer bestimmten Anweisung 1000 mal berechnet und in einer Datei dmy.dat gespeichert.

```
LOOP [varname=n TO m [BY {1**}]]
                        {n  }
     [IF [()logical expression()]]

Transformationen

END LOOP [IF [()logical expression()]]
```

6.4.4 Umcodieren einzelner Werte einer Variablen:

➔ **recode, recod**

Will man bestimmte Werte einer oder mehrerer numerischer Variablen umbenennen so verwendet man den Befehl ‘recode’. Es ist auch möglich, umbenannte Werte als eine neue Variable zu schreiben.

```
recod sex (1 eq 0) (2 eq 1)
```

ändert 1 in 0 und 2 in 1

```
recod age (low thr 25 eq 1) (25 thr hig eq 2)
```

ändert alle Werte der Variablen ‘age’, die kleiner als 25 sind in 1, jene, die größer als 25 sind, in 2. Man könnte auch schreiben

```
recod age (low thru 25 eq 1) (else eq 2)
```

```
RECODE Varlist (Werteliste=Wert)
     [(Werteliste=Wert)...]
     [INTO Varlist]
  [/Varlist...]

Eingabe-Schlüsselwörter:
LO, LOWEST, HI, HIGHEST, THRU,
MISSING, SYSMIS, ELSE
```

6.4.5 Bedingte Auswertung:

➔ select if, sel if

Sollen nur bestimmte Werte einer Variablen oder bestimmte Kombinationen von Werten zur Berechnung herangezogen werden, so ist ‘select if’ zu verwenden.

select if sex eq 1
bezieht nur männliche Versuchspersonen in die Berechnung mit ein,
select if (sex eq 1 and age lt 30)
es werden nur Daten von Männern, die jünger als 30 Jahre sind verwendet.

```
SELECT IF [(]logischer Ausdruck[)]
```

6.4.6 Auswahl einer gewissen Anzahl von Fällen:

➔ n of cases, n of cas

Ähnlich wie ‘select if’, dient auch ‘n of cases’ zur Datenfilterung’. Es wird angegeben, nur eine gewissen Anzahl von Fällen in die Berechnung mit aufzunehmen.

n of cases 20
nur die ersten 20 Fälle werden berücksichjtigt. Freilich kann man diese Befehle auch kombinieren
select if (sex eq 2 and age ge 30)
n of cases 10
hier werden die ersten 10 Frauen, die älter als 29 Jahre sind, zur Analyse herangezogen.

```
N OF CASES n
```

6.4.7 Auf eine Prozedur beschränkte Auswahl:

➔ temporary, tem

Will man nur eine Berechnung mit gefilterten Datensätzen durchführen, soll also die Selektion durch die Befehle ‘select if’ und ‘n of cases’ nicht global erfolgen, so schreibt man ‘temporary’ vor die Filterbefehle.

```
temporary
select if sex eq 1
```

```
TEMPORARY
```

6.4.8 Rohdaten sortieren:

➔ **sort cases, sor cas**

Die Rohdaten werden nach einer oder mehreren Variablen geordnet. Dies ist besonders in Verbindung mit dem ‘split file’ Befehl (s. u.) von Bedeutung.

```
sor cas by sex.
```

```
SORT CASES [BY] varlist[({A})]
                        {D}
              [ varlist...]
```

6.4.9 Aufteilung der Rohdatenmatrix:

➔ **split file, spl fil**

Mit ‘split file’ wird die Rohdatendatei anhand einer oder mehrerer Variablen bzw. Variablenkombinationen geteilt, und für jeden Teil eine getrennte Berechnung durchgeführt. Dazu ist es erforderlich, daß die Daten nach der Variablen, nach der die Daten aufgeteilt werden sollen, geordnet sind. Will man beispielsweise getrennte Berechnungen für Männer und Frauen, so teilt man die Daten anhand der Variablen ‘sex’.

```
split file by sex
split file off
```
hebt die Teilung auf, es wird wieder mit dem gesamten Datensatz gearbeitet.

```
SPLIT FILE {BY Varlist}
           {OFF       }
```

6.4.10 Transponierung der Rohdatenmatrix:

➔ **flip, fli**

Durch ‘flip’ wird die Rohdatenmatrix transponiert. Aus Variablen werden Fälle, aus Fällen Variablen. Zweimal ‘flip’ ergibt wiederum die Ausgangsmatrix.

flip all
Die gesamte Rohdatenmatrix wird transponiert.

```
FLIP [[VARIABLES=] {ALL    }]
                   {Varlist}
     [/NEWNAMES=Variable]
```

6.5 GESTALTUNG DER AUSGABEDATEI

6.5.1 Setzen von globalen Eigenschaften:

➔ **set**

‘Set’ bestimmt gewisse Eigenschaften des SPSS-Systems, wie Format des Outputs, Umfang der Ausgabe, Grafikauflösung etc. Alle Einstellungen lassen sich durch **‘show all’** abrufen.

set width 80 lenght none
setzt den Output auf 80 Zeichen Breite und unendliche Länge, also keine Seitenumbrüche,
set highres off
stellt bei SPSS den hochauflösenden Grafikmodus ab. Grafiken werden im Output im ASCII Format dargestellt.

```
SET [ERRORS={ON }]
            {OFF}
[HEADERS={NO   }]
         {YES  }
         {BLANK}
[HIGHRES={ON }]
         {OFF}
[LENGTH={59  }]
        {n   }
        {NONE}
 [WIDTH={132}]
        {n  }
```

6.5.2 Titel in der Ausgabedatei:

➔ **title, tit ; subtitle, sub**

So man will, kann man einen Titel bzw. Untertitel für die Ausgabedatei bestimmen.

```
title 'Meine erste SPSS Berechnung'
subtitle 'SPSS macht SPaSS'
```

```
TITLE ['']Text['']

SUBTITLE ['']Text['']
```

6.5.3 Beschreibung von Variablen:

➔ **variable labels, var lab**

Will man den Variablen einen längeren oder aus mehreren Worten bestehenden Namen geben, so benutzt man 'variable labels'. Diese Beschreibung der Variablen erscheint dann im Output und macht diesen leichter lesbar.

```
variable labels sex 'Geschlecht der Versuchspersonen'
                /age 'Alter der Versuchspersonen'
```

```
VARIABLE LABELS Varname 'Label'
    [/Varname...]
```

6.5.4 Beschreibung einzelner Werte:

➔ **value labels, val lab**

Gleiches wie mit Variablen kann auch mit einzelnen Werten oder 'Skalaren' geschehen. Der Befehl 'value labels' weist einzelnen Werten bestimmte Bezeichnungen zu. Beispielsweise allen Einsern den Namen 'männlich', allen Zweiern den Namen 'weiblich'.

```
value labels sex 1 'maennlich' 2 'weiblich'
```

```
VALUE LABELS Varlist
   Wert 'Label' Wert 'Label'...
 [/Varlist...]
```

6.6 STATISTIKPROZEDUREN

6.6.1 Deskriptive Statistik

6.6.1.1 *Auflisten von Variablen:*

➔ **list, lis**

Mit ‘list’ werden alle Variablen und Fälle, so wie sie definiert wurden, aufgelistet. Siehe einführendes Beispiel.

list all

```
LIST [[VARIABLES=]{ALL**  }]
                  {Varlist}
 [/FORMAT=[{WRAP**}] [{UNNUMBERED**}]]
          {SINGLE}   {NUMBERED    }
 [/CASES=[FROM {1**}][TO {eof**}][BY {1**}]]
               {n  }     {n    }     {n  }
```

6.6.1.2 *Häufigkeitstabellen:*

➔ **frequencies, fre**

Mit dem Befehl ‘frequencies’ wird eine Häufigkeitstabelle der Werte der definierten Variablen erzeugt. Zusätzlich können noch Grafiken und eine Reihe deskriptiver Kennwerte erstellt werden.

frequencies all /statistics mean stddev /histogram

```
FREQUENCIES [VARIABLES=]Varlist[(min,max)]
                        [Varlist...]
 [/FORMAT=[{CONDENSE}][{NOTABLE }]
          {ONEPAGE }  {LIMIT(n)}
 [/HISTOGRAM=[MIN(n)][MAX(n)][{FREQ(n)    }]
                              {PERCENT(n)}
             [{NONORMAL}][INCREMENT(n)]]

              {NORMAL  }
[/STATISTICS=[DEFAULT][MEAN][STDDEV][SUM]
              [MINIMUM][MAXIMUM][RANGE]
              [SEMEAN][VARIANCE][SKEWNESS][SESKEW]
              [MODE][KURTOSIS][SEKURT][MEDIAN]
              [ALL][NONE]]
```

6.6.1.3 ***Kreuztabulation:***

➔ **crosstabs, cro**

'Crosstabs' erstellt Kontingenztabellen, das heißt, die Häufigkeit des Auftretens von bestimmten Wertkombinationen. Diese Prozedur wird häufig zur mehrdimensionalen χ^2 Berechnung herangezogen.

crosstabs sex by age

```
CROSSTABS [TABLES=]Varlist BY Varlist [BY...]
                  [/Varlist...]
 [/MISSING={TABLE**}]
          {INCLUDE}
[/CELLS=[COUNT**][ROW    ][EXPECTED][SRESID ]]
                 [COLUMN][RESID   ][ASRESID]
                 [TOTAL ][ALL     ][NONE    ]

 [/STATISTICS=[CHISQ][LAMBDA][BTAU][GAMMA][ETA ]]
              [PHI  ][UC    ][CTAU][D    ][CORR]
              [CC   ][KAPPA ][RISK][ALL  ][NONE]
```

6.6.1.4 ***Regressionsgraph:***

➔ **plot, plo**

'Plot' zeichnet einen Regressionsgraphen. Diese Prozedur kann man einsetzen, wenn man Zusammenhänge zwischen Variablen bildlich darstellen möchte.

plot /plot schuh with age

```
PLOT [HSIZE={80**}] [/VSIZE={40**}]
            {n   }          {n   }
[/TITLE='Titel']
/PLOT={Varlist} WITH Varlist [(PAIR)]
       {ALL    }
      [BY Varname]
      [;Varlist...]
 [/PLOT=...]
```

6.6.1.5 ***Explorative Datenanalyse:***

➔ **examine, exa**

Für weitreichendere Deskriptionen von Variablen sollte dieser Befehl verwendet werden. 'Examine' erlaubt auch Vergleiche von Untergruppen, mehrere graphische Aufbereitungen der Daten, Prüfung auf Normalverteilung von Daten, Identifikation von Ausreißern etc.

```
examine schuh iq by age sex
  /plot boxplot npplot
```

```
EXAMINE VARIABLES=Varlist [[BY Varlist] [Varname BY Varname]]
[/FREQUENCIES [FROM(Anfangswert)] [BY(Inkrement)]]
[/PLOT=[STEMLEAF**] [BOXPLOT**] [NPPLOT]]

        [SPREADLEVEL(Wert)] [HISTOGRAM]
        [{ALL }]
         {NONE}
 [/STATISTICS=[DESCRIPTIVES**] [EXTREME({5})]]
                                       {n}
              [{ALL }]
               {NONE}
```

6.6.2 Inferentielle Statistik

6.6.2.1 *Verbundenes und unverbundenes 2-Stichprobenproblem (abhängiger und unabhängiger t-Test):*

➔ **t-test, t-t**

Die Prozedur 't-test' berechnet den abhängigen oder unabhängigen t-Test.

⌨
t-test groups=sex(1 2) /variables schuh
berechnet den unabhängigen t-Test bezüglich Schuhgröße zwischen Männern und Frauen,
t-test pairs=iq1 with iq2
berechnet den abhängigen t-Test zwischen IQ1 und IQ2.

🗐

```
Unabhängige Stichproben:
T-TEST GROUPS=Varname({1,2**      })
                      {Wert      }
                      {Wert,Wert}
 /VARIABLES=Varlist
[/CRITERIA=CI[{(0.95 )}]]
               {(Wert)}

Abhängige Stichproben:
T-TEST PAIRS=Varlist[WITH Varlist
           [(PAIRED)]]
           [/Varlist...]
 [/CRITERIA=CI[{(0.95 )}]]
               {(Wert)}
```

6.6.2.2 *Unverbundenes k-Stichprobenproblem (einfaktorielle unabhängige Varianzanalyse):*

➔ **oneway, one**

Mit dem Befehl 'oneway' wird die einfaktorielle unabhängige Varianzanalyse berechnet.

⌨ **oneway schuh by age(1 3)**

```
ONEWAY  Varlist BY Varname(min,max)
[/CONTRAST=Koeffizientenliste]
 [/CONTRAST=... ]
 [/RANGES={LSD          }({0.05 })]
          {DUNCAN       } {alpha}
          {SNK          }
          {TUKEYB       }
          {TUKEY        }
          {MODLSD       }
          {SCHEFFE      }
          {Bereichswerte}
 [/RANGES=...]
[/STATISTICS=[NONE      **]]

              [DESCRIPTIVES]
              [EFFECTS     ]
              [HOMOGENEITY ]
              [ALL         ]
```

6.6.2.3 *Faktorielle Pläne (mehrfaktorielle Varianzanalyse, ANOVA):*

➔ **anova, ano**

'Anova' berechnet eine mehrfaktorielle univariate Varianzanalyse. Es ist auch möglich eine einfaktorielle univariate Varianzanalyse, wie mit 'oneway' zu definieren.

anova schuh by age(1 3)

```
ANOVA [VARIABLES=] Varlist BY Varlist(min,max)
                              Varlist(min,max)
                           [WITH Varlist]
 [/COVARIATES={FIRST**}]
              {WITH   }
              {AFTER  }
 [/MAXORDERS={ALL** }]
             {n     }
             {NONE  }
 [/METHOD={UNIQUE**    }]
          {EXPERIMENTAL}
          {HIERARCHICAL}
 [/STATISTICS=[MCA*][REG*][MEAN*][ALL][NONE]]
```

6.6.2.4 *Faktorielle Pläne, multivariater Fall (Abhängige Varianzanalyse):*

➔ manova, man

Die multivariate ANOVA (MANOVA) ist für alle Arten der Varianzanalyse geeignet. Mit dieser Prozedur können zusätzlich noch mehrere abhängige Variablen (AV) in das Design aufgenommen werden. Da eine AV zu mehreren Zeiten gemessen wie unterschiedliche AVs gehandelt wird, wird die 'manova' Prozedur auch für Varianzanalysen mit Meßwiederholung eingesetzt.

manova iq1 iq2 iq3 /wsfactors=geist (3) /design

```
MANOVA abhängige Varlist [BY unabh. Variablen (min,max)
              [unabh. Variablen...] [WITH Kovariatenliste]]
 [/WSFACTORS=Name (Niveaus) Name...]
[/WSDESIGN=Effekt Effekt...]

[/{PRINT  }=
   {NOPRINT}
      [CELLINFO ([MEANS][SSCP][COV][COR][ALL])]
      [HOMOGENEITY ([BARTLETT][COCHRAN][BOXM][ALL])]
      [DESIGN ([ONEWAY][OVERALL][DECOMP]
               [BIAS][SOLUTION][REDUNDANCY]
               [COLLINEARITY][ALL])]
[/DESIGN={[CONSTANT...]                          }]
         {[Effekte BY Effekte...]                 }
         {[Effekte {WITHIN} Effekte...]           }
```

6.6.2.5 *Nonparametrische Testverfahren:*

➔ npar tests, npa tes

'Npar tests' stellt eine Reihe von nonparametrischen Testverfahren zur Verfügung.

npar tests m-w=note by sex (1 2)
berechnet den Mann-Whitney U-Test
/wilcoxon iq1 with iq2
berechnet den Wilcoxon-Test
/chisquare sex /expecte equal
berechnet einen eindimensionalen CHI-Quadrat Test.

```
NPAR TESTS [CHISQUARE=Varlist[(lo,hi)]/]
           [/EXPECTED={EQUAL      }]
                      {f1,f2,...fn}
 [/K-S({UNIFORM[,lo,hi]})=Varlist]
       {NORMAL[,m,sd]  }
       {POISSON[,m]    }
[/MCNEMAR=Varlist [WITH Varlist [(PAIRED)]]]
 [/SIGN=Varlist [WITH Varlist [(PAIRED)]]]
 [/WILCOXON=Varlist [WITH Varlist [(PAIRED)]]]

[/M-W=Varlist BY Var (v1,v2)]
[/K-W=Varlist BY Var (v1,v2)]
 [/STATISTICS=[DESCRIPTIVES][QUARTILES][ALL]]
```

6.6.2.6 ***Bivariate Korrelation:***

➔ **correlations, cor**

Mit dieser Prozedur wird die Stärke einer linearen statistischen Beziehung zwischen zwei oder mehreren intervallskalierten Merkmalen über den Pearsonschen Korrelationskoeffizient berechnet.

correlations schuh with Iq

```
CORRELATIONS [VARIABLES=] Varlist [WITH Varlist]
                          [/Varlist...]
[/PRINT={TWOTAIL**}  {SIG**}]
        {ONETAIL  }  {NOSIG}
 [/FORMAT={MATRIX**}]
          {SERIAL  }
[/STATISTICS=[DESCRIPTIVES][XPROD][ALL]]
```

6.6.2.7 ***Bivariate Korrelation höherer Ordnung (Partialkorrelation):***

➔ **partial corr, par cor**

'Partial corr' berechnet die Korrelation zweier oder mehrerer Variablen unter Herauspartialisierung einer oder mehrerer Variablen.

Partial corr schuh with iq by age

```
PARTIAL CORR [VARIABLES=] Varlist
  [WITH Varlist]
   BY Steuerliste (Niveaus)
[/Varlist...]
[/SIGNIFICANCE={TWOTAIL**}]
               {ONETAIL  }
[/STATISTICS=[NONE**][CORR]   ]
  [DESCRIPTIVES][BADCORR][ALL]
```

6.6.2.8 ***Multiple Regressionsanalyse:***

➔ **regression, reg**

'Regression' errechnet den Zusammenhang einer AV mit mehreren UVs. Sehr hilfreich ist diese Prozedur bei der Konstruktion von Einzelvergleichen und Kontrasten in mehrfaktoriellen Designs, was ausführlich bei Bortz (1993) beschrieben wird.

Regression /var schuh age iq /dep schuh /met ent age iq..

```
REGRESSION
  [/VARIABLES={Varlist    }]
              {(COLLECT)**}
              {ALL        }
             {n    }
[/STATISTICS=[DEFAULTS**][R**][COEFF**]
               [ANOVA**][OUTS**]
               [ZPP][CHA][CI][F][BCOV]
               [SES][LINE][HISTORY][XTX][COLLIN]
               [END][TOL][SELECTION][ALL]]
/DEPENDENT=Varlist
  [/METHOD=]{STEPWISE [Varlist]       }
            {FORWARD [Varlist]        }
            {BACKWARD [Varlist]       }
            {ENTER [Varlist]          }
            {REMOVE Varlist           }
            {TEST(Varlist)(Varlist)...}
[/SCATTERPLOT=[SIZE({SMALL})](Varname,Varname)...]
                      {LARGE}
```

6.6.2.9 ***Faktorenanalyse:***

➔ **factor, fac**

'Factor', bei kleineren Datenmengen zur (vorsichtigen) Theorienbildung. Bei großen Datenmengen zu deren Reduktion, in der psychologischen Forschung jedoch häufiger zur Hypothesenprüfung und Theorienbildung.

```
Factor var all.
```

```
FACTOR VARIABLES=Varlist***
[/PLOT=[EIGEN] [ROTATION [(n1,n2)]]]
[/FORMAT=[SORT] [BLANK(n)] [DEFAULT**]]
 [/CRITERIA=[FACTORS(n)] [MINEIGEN({1.0**})]
                                  {eig  }
            [ITERATE({25**})] [{KAISER**}]
                      {ni  }   [{NOKAISER}]
            [RCONVERGE({0.0001**})] [DELTA({0**})]
                       {r1      }          {d  }

            [ECONVERGE({0.001**})]] [DEFAULT**]
                       {e1     }
```

6.7 MAKRO-PROGRAMMIERUNG:

➔ **define; !enddefine**

Sollen beispielsweise gewisse Prozeduren wiederholt für unterschiedliche Datensätze oder Variablen durchgeführt werden, so werden diese Prozeduren in eine Makro-Struktur (ein Unterprogramm) eingebettet. Dieses Makro erhält einen bestimmten Namen und kann durch diesen Namen aufgerufen werden. Zusätzlich können unterschiedliche Argumente (Variablen) mit dem Aufruf an die Makro-Struktur übergeben werden. Makros können auch in Kombination mit Schleifen (loop) und bedingten Transformationen ('if' bzw. 'do if') eingesetzt werden. Dies kann mitunter zu erheblichen Kürzungen von SPSS-Programmskripts führen.

```
define !makro (!pos !tok (1)).
t-test gro=gr /var !1.
!enddefine.
!makro iq.
```

Es wird mit 'define' ein Makro mit dem Namen '*makro'* definiert. Es kann ein Argument aufgenommen werden '(!pos !tok (1))'. Die unabhängige t-Test Prozedur wird in das Makro eingebettet. Mit '`!makro`' wird das Makro aufgerufen und die Variable '`iq`'

übergeben. ‘Iq’ wird nun an alle Stellen des Makros gesetzt an denen der Platzhalter ‘!1’ (bedeutet Argument an der erster Position) steht.

```
DEFINE !macroname([{argument name=}{!TOKENS (n)          }])
                  {!POSITIONAL=  }{!CHAREND('char')       }
                                  {!ENCLOSE('sym1','sym2')}
                                  {!CMDEND                }
    [!DEFAULT (default string)] [!NOEXPAND]
    [/{argument name=} ...] ] )
      {!POSITIONAL=  }

macro body

!ENDDEFINE
```

LITERATUR

Allerbeck, K. (1977). Computerunterstützte Datenaufbereitung und Datenanalyse. In: Koolwijk, J. v. & M. Wieken - Maysen (hg.), Bd. 7, 170-188.
Barthel, R. (1990). Das VMS Buch. Sybex, Düsseldorf.
Böhning, D. & Wilke, H. (1980). Statistische Analysepakete und ihre Einbeziehung in sozialwissenschaftliche Studiengänge. Zeitschrift für Soziologie. 9, 290-307.
Bortz, J. (1993). Statistik für Sozialwissenschaftler. Springer, Berlin.
Kähler, W. M. (1994). SPSS für Windows; Datenanalyse unter Windows.Vieweg, Braunschweig/ Wiesbaden.
Wittenberg, R. (1991). Computerunterstützte Datenanalyse. Gustav Fischer, Stuttgart.
Wittenberg, R., Cramer, H. (1992). Datenanalyse mit SPSS. Gustav Fischer, Stuttgart, Jena.

INTERNET

http://bkfug.kfunigraz.ac.at/~statist/stats.htm
http://www.spss.com